JUL 10 2006

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	6.Jul.06	THESIS

4. TITLE AND SUBTITLE
UAV MISSION PLANNING UNDER UNCERTAINTY.

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT SAKAMOTO PHILEMON

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI04-1819

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES
209

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

UAV Mission Planning Under Uncertainty

by

Philemon Sakamoto
B.S. Operations Research, B.S. Economics
United States Air Force Academy, 2004

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of
Masters of Science in Operations Research
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2006

Author .....
Sloan School of Management
Interdepartmental Program in Operations Research
May 18, 2006

Certified by .....
Stephan E. Kolitz
The Charles Stark Draper Laboratory
Technical Supervisor

Certified by .....
Cynthia Barnhart
Professor, Civil and Environmental Engineering Department
Engineering Systems Division
Thesis Advisor

Accepted by .....
James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Codirector, Operations Research Center

UAV Mission Planning Under Uncertainty

by

Philemon Sakamoto

Submitted to the Sloan School of Management
on May 18, 2006, in partial fulfillment of the
requirements for the degree of
Masters of Science in Operations Research

Abstract

With the continued development of high endurance Unmanned Aerial Vehicles (UAV) and Unmanned Combat Aerial Vehicles (UCAV) that are capable of performing autonomous functions across the spectrum of military operations, one can envision a future military in which Air Component Commanders control forces comprised exclusively of unmanned vehicles. In order to properly manage and fully realize the capabilities of this UAV force, a control system must be in place that directs UAVs to targets and coordinates missions in a manner that provides an efficient allocation of resources. Additionally, a mission planner should account for the uncertainty inherent in the operations. Uncertainty, or stochasticity, manifests itself in most operations known to man. In the battlefield, such unknowns are especially real; the phenomenon is known as the *fog of war*.

A good planner should develop plans that provide an efficient allocation of resources and take advantage of the system's true potential, while still providing ample "robustness" in plans so that they are more likely executable and for a longer period of time.

In this research, we develop a UAV Mission Planner that couples the scheduling of tasks with the assignment of these tasks to UAVs, while maintaining the characteristics of longevity and efficiency in its plans. The planner is formulated as a Mixed Integer Program (MIP) that incorporates the Robust Optimization technique proposed by Bertsimas and Sim [12].

Technical Advisor: Andrew P. Armacost, Lt Col, USAF
Associate Professor and Director of Operations Research
United States Air Force Academy

Thesis Supervisor: Stephan E. Kolitz
The Charles Stark Draper Laboratory

Thesis Advisor: Cynthia Barnhart
Professor, Civil and Environmental Engineering Department
Engineering Systems Division

UAV Mission Planning Under Uncertainty

by

Philemon Sakamoto

Submitted to the Sloan School of Management
on May 18, 2006, in partial fulfillment of the
requirements for the degree of
Masters of Science in Operations Research

Abstract

With the continued development of high endurance Unmanned Aerial Vehicles (UAV) and Unmanned Combat Aerial Vehicles (UCAV) that are capable of performing autonomous functions across the spectrum of military operations, one can envision a future military in which Air Component Commanders control forces comprised exclusively of unmanned vehicles. In order to properly manage and fully realize the capabilities of this UAV force, a control system must be in place that directs UAVs to targets and coordinates missions in a manner that provides an efficient allocation of resources. Additionally, a mission planner should account for the uncertainty inherent in the operations. Uncertainty, or stochasticity, manifests itself in most operations known to man. In the battlefield, such unknowns are especially real; the phenomenon is known as the *fog of war*.

A good planner should develop plans that provide an efficient allocation of resources and take advantage of the system's true potential, while still providing ample "robustness" in plans so that they are more likely executable and for a longer period of time.

In this research, we develop a UAV Mission Planner that couples the scheduling of tasks with the assignment of these tasks to UAVs, while maintaining the characteristics of longevity and efficiency in its plans. The planner is formulated as a Mixed Integer Program (MIP) that incorporates the Robust Optimization technique proposed by Bertsimas and Sim [12].

Technical Advisor: Andrew P. Armacost, Lt Col, USAF
Associate Professor and Director of Operations Research
United States Air Force Academy

Thesis Supervisor: Stephan E. Kolitz
The Charles Stark Draper Laboratory

Thesis Advisor: Cynthia Barnhart
Professor, Civil and Environmental Engineering Department
Engineering Systems Division

Acknowledgments

I am thankful for the support and guidance of all those involved in my research. I thank Steve Kolitz, for his guidance throughout the process. I thank Cindy Barnhart, for the helpful insight she provided throughout our many meetings. Thank you, Lt Col Armacost, for devoting your time to the research- your input was a real help for me. I thank Francis Carr and Ramsay Key for the technical support and mentorship that they provided me with. Thank you, "Doc" Morgan and Col. "Stumper" Stimpson, for being such invaluable repositories of AOC knowledge. I thank my family for their never-failing love and encouragement. I am thankful to my AF friends, who stuck together through thick and thin.

I extend my deepest thanks and gratitude to the many brothers and sisters in the Lord who shepherded me and cared for me these past two years. I can truly say that I've experienced the love that is in the body of Christ. To have so many companions with whom to run the race was both a great salvation for me and a constant source of encouragement. My experiences with you made these past two years meaningful, and I feel that my time here was witness to a building work and to growth in life that had eternal value. You will always be in my thoughts and prayers.

I thank the *God and Father of all, who is over all, and through all, and in all.* It is by His sovereign arrangement that I could be here, and I am persuaded that He will continue to keep and guard me wherever I go. 1 Thess 5:24- *Faithful is He who calls you, who also will do it.*

The thesis was prepared at The Charles Stark Draper Laboratory, Inc. under Internal Company Sponsored Research Project 20442-001.

Philemon Sakamoto May 18, 2006

Contents

1 Introduction **17**

1.1 Thesis Overview 18

1.2 Contributions 19

2 The Targeting Process and the Air Operations Center **21**

2.1 The Joint Targeting Cycle 22

2.1.1 Phase 1 - Commander's Objectives, Guidance, and Intent . . . 23

2.1.2 Phase 2 - Target Development, Validation, Nomination, and Prioritization 24

2.1.3 Phase 3 - Capabilities Analysis 24

2.1.4 Phase 4 - Commander's Decision and Force Assignment 25

2.1.5 Phase 5 - Mission Planning and Force Execution 25

2.1.6 Phase 6 - Combat Assessment 26

2.2 The Joint Air Operations Center 27

2.2.1 Operational Mapping of Functional Phases 27

2.2.2 The Strategy Division 31

2.2.3 The Combat Plans Division 31

2.2.4 The Combat Operations Division 32

2.2.5 The ISR Division 32

2.2.6 The Air Mobility Division 33

2.2.7 The Final Link - Operational Units 33

3 Model Development **35**

3.1 Scope 36

3.2 The UAVMPP 39

3.2.1 Inputs 40

3.2.2 Outputs 41

3.2.3 Assumptions 42

3.3 Summary 43

4 Problem Formulation **45**

4.1 Linear Programming 46

4.2 Integer Programming 48

4.2.1 The Vehicle Routing Problem 49

4.2.2 Methodology 52

4.2.3 Discussion 60

4.3 Robust Optimization 60

4.3.1 The Bertsimas-Sim formulation 62

4.4 UAV Mission Planning Problem Formulation 65

5 Tests and Analysis **75**

5.1 Metrics 75

5.2 Test Bed 80

5.2.1 Simulation Architecture 85

5.2.2 Scenarios 86

5.3 Hypotheses 88

5.3.1 Beliefs on Performance 89

5.4 Tests and Results 95

5.4.1 Experiment 1 95

5.4.2 Calibration 113

5.4.3 Experiment 2 114

5.4.4 Experiment 3 137

5.4.5 Experiment 4 144

5.4.6 Experiment 5 152
5.4.7 Experiment 6 164
5.4.8 Experiment 7 181
5.5 Summary 187

6 Conclusion **193**
6.1 Usefulness of Protection 193
6.2 The Effectiveness of the Bertsimas-Sim formulation 194
6.3 Recommendations on Modification 196
6.4 Future Work 199

A "Value of Information" Formulation **201**

B "ISR before Strike" Formulation **203**

List of Figures

2-1 The Joint Targeting Cycle, as presented by [20] 23

2-2 A representation of the Combat Assessment components, as presented by [20] 27

2-3 A functional to operational mapping of the targeting cycle, as presented by [20] 28

2-4 A hierarchical view of the Theater Air Control System 29

2-5 The Air Operations Center with its Divisions 30

3-1 The subdivision of the Targeting Cycle by decision-levels 37

3-2 The operational mapping of the decision-levels 37

4-1 a graphical representation of a linear programming problem's feasible region 48

4-2 A network representation of a dynamic programming problem 54

4-3 an example implementation of stochastic Dynamic Programming . . . 55

4-4 An example of a Branch-and-Bound algorithm, adapted from [13] . . 57

4-5 Robust Optimization considers only a subset of the feasible region . . 61

5-1 Overview of simulation with parameter flow 81

5-2 Simulation test bed 82

5-3 Simulation architecture for Nominal Sim 83

5-4 Simulation architecture for Smart Sim 84

5-5 Example map of scenarios 88

5-6 Increasing $\hat{x}$ 93

5-7 The effect of flattening the distribution of uncertainty 93
5-8 Increasing the number of uncertain variables 93
5-9 Map of Scenario 1 96
5-10 Map of Scenario 2 97
5-11 Map of Scenario 3 98
5-12 Map of Scenario 4 98
5-13 Sim and objective function values for Experiment 1, Scenario 1 99
5-14 Experiment 1, Scenario 1 results 101
5-15 Sim and objective function values for Scenario 2 105
5-16 Experiment 1, Scenario 2 results 106
5-17 Runtime and optimality gap for Scenario 2 107
5-18 Sim and objective function values for Scenario 3 108
5-19 Experiment 1, Scenario 3 results 109
5-20 Sim and objective function values for scenario 4 111
5-21 Experiment 1, Scenario 4 results 112
5-22 Map of Base Scenario 114
5-23 Sim and objective function values for Scenario 1 116
5-24 Experiment 2, Scenario 1 results 119
5-25 Sim and objective function values for Scenario 2 121
5-26 Runtime and Optimality Gap for Scenario 2 121
5-27 Experiment 2, Scenario 2 results 124
5-28 Sim and objective function values for Scenario 3 125
5-29 Constraint Violation by Vehicle Type 125
5-30 Experiment 2, Scenario 3 results 127
5-31 Percent change in metrics for Scenarios 1-3 129
5-32 Sim and objective function values for Scenario 4 131
5-33 Experiment 2, Scenario 4 results 132
5-34 Sim and objective function values for Scenario 5 133
5-35 Experiment 2, Scenario 5 results 135
5-36 Percent change in metrics for Scenarios 4 and 5 136

5-37 Sim and objective function values for Scenario 1 138
5-38 Runtime and optimality gap for Scenario 1 138
5-39 Experiment 3, Scenario 1 results 140
5-40 Sim and objective function values for Scenario 2 141
5-41 Experiment 3, Scenario 2 results 142
5-42 Percent change in metrics for Scenarios 1 and 2 143
5-43 Sim and objective function values for Scenario 1 145
5-44 Experiment 4, Scenario 1 results 147
5-45 Sim and objective function values for Scenario 2 148
5-46 Experiment 4, Scenario 2 results 149
5-47 Percent change in metrics for Scenarios 1 and 2 151
5-48 Constraint violation when no uncertainty exists 153
5-49 Sim and objective function values for Scenario 1 153
5-50 Experiment 5, Scenario 1 results 155
5-51 Sim and objective function values for Scenario 2 157
5-52 Experiment 5, Scenario 2 results 158
5-53 Sim and objective function values for Scenario 3 159
5-54 Experiment 5, Scenario 3 results 160
5-55 Sim and objective function values for Scenario 4 161
5-56 Experiment 5, Scenario 4 results 163
5-57 Sim and objective function values when the objective function is protected 166
5-58 Experiment 6, Scenario 1 results 168
5-59 Sim and objective function values when Separation Constraints are protected 170
5-60 Experiment 6, Scenario 2 results 171
5-61 Sim and objective function values when opportunity windows are protected 173
5-62 Experiment 6, Scenario 3 results 174

5-63 Sim and objective function values when Opportunity Window close times are protected 176
5-64 Experiment 6, Scenario 3a results 177
5-65 Sim and objective function values when the Range Constraint is protected 179
5-66 Experiment 6, Scenario 4 results 180
5-67 Sim and objective function values for modified formulation, under the original weighting 182
5-68 Sim and objective function values for both original and modified formulations, under the new weighting 182
5-69 Experiment 7, Scenario 1 results 186

List of Tables

3.1 Comparison of the two UAV types 40

4.1 Elements of the VRPTW 51
4.2 Elements of the UAVMPP 66
4.3 A line by line description of the UAVMPP 67
4.4 Dual constraints of the Robust UAVMPP formulation 71

5.1 The three types of constraints subject to uncertainty 77
5.2 Overview of target types 87
5.3 Scenario 1 Plan Itineraries 120
5.4 Base Hypotheses 188

Chapter 1

Introduction

With the continued development of high endurance Unmanned Aerial Vehicles (UAVs) and Unmanned Combat Aerial Vehicles (UCAV) that are capable of performing autonomous functions across the spectrum of military operations, one can envision a future military in which commanders control forces comprised of unmanned vehicles. These UAVs will work cooperatively to provide Intelligence, Surveillance, and Reconnaissance (ISR), strike, and Battle Damage Assessment (BDA) over the battlefield for extended periods of time. The forces will not be subject to some of the constraints faced by manned aircraft. They are well-suited for dull, dirty and dangerous missions: very long, "dull", missions that prosecute multiple targets and are physically taxing on humans in terms of time; chemical and biological agent sensing missions that are "dirty" and inherently dangerous to humans; and "dangerous" missions into high threat areas. [17]

To properly manage and fully realize the capabilities of this UAV force, a command and control system must be in place that directs UAVs to targets and coordinates missions in a manner that provides an efficient allocation of resources. A mission planner in the command and control system should account for the uncertainty inherent in the operational execution of the missions. Uncertainty, or stochasticity, manifests itself in most operations known to man. Some factors or underlying processes involved in the operations are too complex to include in modeling, and we are compelled to use models that are merely estimates of the operations. Furthermore, these estimates

are often based on data we gather through observation, which are themselves subject to flaws from measurement error and collection. For example, a unit transporting equipment between installations cannot know with certainty the time required to traverse a leg of its route, nor can it know beforehand the fuel it will need for the trip. Its expected fuel consumption is likely to be a data-based estimate, and the time required to navigate the leg is vulnerable to unforeseen events such as engine malfunction or traffic accidents. In the battlefield, such unknowns are even more real; the phenomenon is known as the *fog of war.*

1.1 Thesis Overview

A good planner must develop plans that provide an efficient allocation of resources and take advantage of the system's true potential, while still providing ample "robustness" in plans so that these plans are more likely executable for a longer period of time. The goal of this research is to develop an automated planner for a system of UAVs that allocates system resources efficiently while still producing plans that are executable operationally. We now give a chapter overview of the remainder of the thesis:

Chapter 2 -The Targeting Process. In this chapter we present the Targeting Cycle, which is the paradigm currently used by the US Air Force (USAF) to view the process of finding candidate targets, selecting them for prosecution, assigning them to operational units, and overseeing their prosecution. We also present the Air Operations Center (AOC), which comprises the operational infrastructure that carries out all of the tasks described in the Targeting Cycle. We explain the organization of the AOC and where it fits into the USAF chain of command. While a few UAV squadrons are currently maintained by the USAF, they have not yet developed to the operational state that this research envisions. Nonetheless, the current framework allows us to understand the tasks involved in this problem, gives us the organizational hierarchy with which to carry them out, and will serve as a basis for our modeling of the a system of UAVs.

Chapter 3 -Model Development. This chapter defines the scope of the problem we choose to consider, and discusses its relationship to the framework presented in chapter 2. It establishes the operational framework that we will use to model this UAV Mission Planning Problem (UAVMPP), describes the inputs and outputs of our problem, and explains any assumptions that we make in it.

Chapter 4 -Problem Formulation. In this chapter we develop the mathematical formulation of our problem. We provide a brief discussion of linear programming as a background to the formulation, and explain the general class of Vehicle Routing Problems (VRPs) that our problem belongs to. The chapter gives an overview of the various methods available for solving VRPs such as ours, and provides a literature review of the implementations of these methods, when available. The chapter discusses the field of Robust Optimization, presents a few of the ways in which uncertainty has been incorporated into linear programming, and develops the method suggested by Bertsimas and Sim [12]. We then present our formulation for the UAV mission planner, and explain any assumptions we make in it.

Chapter 5 -Tests and Analysis. This chapter covers the tests we run on our model and reports the results and analysis of the tests. It establishes our objectives in the testing, provides the hypotheses that we test, describes the metrics used to discriminate among plans, and presents the test bed used for experimentation.

Chapter 6 -Conclusion. This chapter summarizes our work and findings. We discuss the effectiveness of protection against uncertainty in our formulation, give suggestions for the modification of our formulation and the Bertsimas-Sim method of protection, and provide suggestions for future work in the field.

1.2 Contributions

We now list the major contributions of this research:

1. A formulation for the UAV Mission Planning Problem that combines the scheduling of tasks with the assignment of these tasks to UAVs. This formulation could serve the role of a mid-level planner for a system of UAV units, if such a system

is realized.

2. The incorporation of the Bertsimas-Sim method of protection from uncertainty into this formulation. The formulation is the first application of the Bertsimas-Sim method to a Vehicle Routing Problem with Time Windows (VRPTW), to our knowledge. The research makes a few simplifications in implementing the method.

3. The development of a set of metrics with which to compare solutions for the UAVMPP. These metrics can be extended for use in other problems, where solutions must be measured not only by their objective function value, but also by their robustness.

4. A test bed for our UAVMPP formulation. The test bed evaluates solutions via simulation, and generates realizations of the uncertain coefficients from the coefficient uncertainty ranges. The test bed can handle different probability distribution functions for the coefficients, and tests the constraints for feasibility in a manner that accounts for constraint dependencies.

5. A "Smart Sim", that approximates an intelligent controller, and avoids unreachable targets and overextending UAV range.

6. In depth experimentation of the formulation. The testing provides insight into the relative performance of protected solutions to those that are not protected from uncertainty. The testing also provides insight into the quality of the Bertsimas-Sim method of protection when applied to a VRP-type problem.

7. A modified UAVMPP formulation that addresses some of the problems discovered in the previous formulation. The modification involves the introduction of constraints in the objective function as Lagrangians. The resulting formulation maximizes slack in the constraints, in addition to optimizing objective function value, and behaves more predictably with Bertsimas-Sim protection. The modification can be applied to other problems where solutions must be differentiated on the basis of robustness.

Chapter 2

The Targeting Process and the Air Operations Center

In this chapter we present functional and operational views of the targeting process, as developed by the United States military. We present the process in its functional form as a cycle that can be repeated over and over in a given conflict. Because of this, the process is referred to as the "Joint Targeting Cycle" (JTC) in the governing military doctrine publication, JP 3-60 [20]. The moniker, "joint," is added to indicate that the cycle pertains not only to air operations, but also to land and naval operations. We continue by mapping this functional framework to an operational one, and explain how the functions defined in the Joint Targeting Cycle are accomplished operationally, and which operational bodies are responsible for their accomplishment. In particular, we present the Joint Air Operations Center (JAOC) as the governing body that plans and executes the phases of the targeting process from inception to completion. Finally, we add a discussion on the task of matching the missions created by the JAOC to specific pilots and aircraft, as this task is not explicitly covered in targeting doctrine.

This section on current practices will serve as a background for our problem, the UAV Mission Planning Problem (UAVMPP), and a basis for the model that we develop for it. As mentioned earlier, our UAV force is merely conceptual and does not exist operationally to date. Because of this, some might raise the argument that,

even if this force were to be realized, discussion on current targeting practices is not relevant, as the proposed planner is designed for a system that is different from anything in existence and should operate under a system designed independently from current practice. For this reason, we feel it is necessary to establish the relevence of this discussion to the UAVMPP. We first claim that while hypothetical, a fighting unit composed entirely of unmanned vehicles is not far-fetched. Such a system rests within the military's senior leadership's vision for the USAF [17], and the USAF is actively supporting programs to further the already rapidly developing UAV capabilities. A squadron of remotely controlled UAVs is already maintained by the USAF, and current research continues to push the boundaries of vehicle autonomy. Secondly, when the USAF does realize such a system, it will integrate it within the targeting framework already existing in the USAF. While the unit, comprised now of unmanned vehicles, will exhibit some operational differences from units currently in existence, it will be designed to serve the same functions as these units, and should therefore possess many similarities.

2.1 The Joint Targeting Cycle

The Joint Targeting Cycle consists of 6 phases (see figure 2-1):

1. Commander's Objectives, Guidance, and Intent
2. Target Development, Validation, Nomination, and Prioritization
3. Capabilities Analysis
4. Commander's Decision and Force Assignment
5. Mission Planning and Force Execution
6. Combat Assessment

While each phase appears sequentially in the cycle, the phases are not time dependent, and can occur simultaneously. We describe each phase and the key tasks and players within each

2.1.1 Phase 1 - Commander's Objectives, Guidance, and Intent

The commander's objectives and guidance are established during this phase. Commander's objectives support the National Command Authority (NCA) desired end state, and the commander's guidance stipulates particular conditions, such as limitations on collateral damage, related to military operations in the conflict. When taken together, the objectives and guidance form the commander's intent. The focus of this intent should always be to affect some change in the enemy's actions or behavior.

Phase 1 drives the entire operation, and the success of this phase determines the overall effectiveness of the next phases. Main tasks during this phase are to translate national strategy into clear and quantifiable tasks, and to develop Measures of Effectiveness (MOEs) that can be used to determine the degree to which the objectives and NCA goals are accomplished.

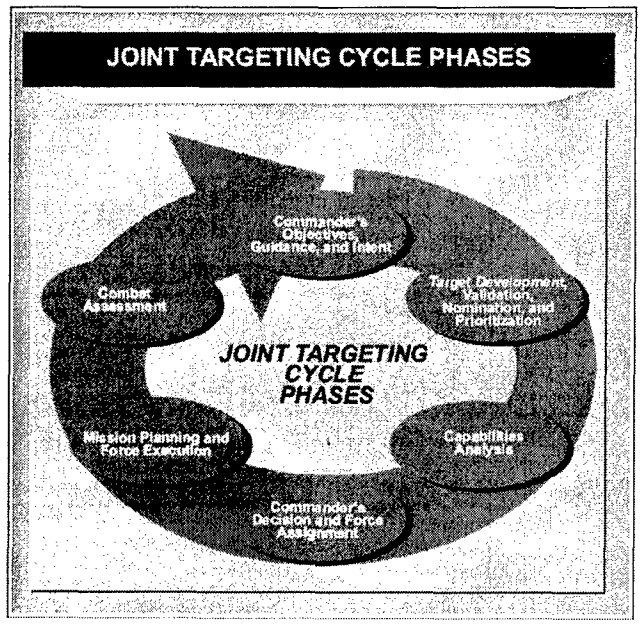


Figure 2-1: The Joint Targeting Cycle, as presented by [20]

2.1.2 Phase 2 - Target Development, Validation, Nomination, and Prioritization

In Phase 2 of the Targeting Cycle, a list of potential targets is developed, and the targets are validated, nominated, and prioritized. Commander's objectives are normally directed at enemy capabilities. These capabilities are made possible through physical and virtual infrastructures maintained by the enemy. The steps in Phase 2 involve mapping physical targets to the capabilities at which the objectives of Phase 1 are directed. These targets can be thought of as occuring within systems: targets that are related geographically and function together as a unit to achieve a particular capability. These systems are intra- and interdependent, and the key to good targeting is to recognize these dependencies and develop target sets that address the capabilities of interest in the most efficient manner and capitalize on enemy vulnerabilities. Some targets, for instance, might be involved in multiple capabilities of interest, and destroying them could help to achieve multiple objectives. Furthermore, destroying one critical target might be as effective in halting an enemy capability as destroying several less-important ones.

Cooperation among planners and intelligence experts is important to good target development. The intelligence community tasks its theater assets using input from planners in order to provide the planners with information on targets from the systems that they are interested in affecting. These targets are then validated with up-to-date information to ensure that they are still used by the enemy, that they are lawful according to the Laws of Armed Conflict (LOAC), and consistent with rules of engagement (ROE). Following validation, targets that pass the criteria are "nominated" and placed onto a Joint Integrated Prioritized Target List (JIPTL).

2.1.3 Phase 3 - Capabilities Analysis

This phase of targeting begins the process of assigning the already-determined targets to the assets and capabilities of the friendly forces. Before targets can be assigned to friendly assets, an analysis of the assets and capabilities of friendly forces must first

take place. From an Air Force perspective, things that might be considered include the aircraft available for use within the theater, the munitions that each can carry, and a measure of the potential effectiveness of these munitions and ordnances against the targets specified in the target list. Such analysis gives decision-makers an idea of the number of targets that could be reasonably prosecuted, the time frame in which this can be accomplished, as well as the particular assets that should be assigned to each target to make best use of the assets.

2.1.4 Phase 4 - Commander's Decision and Force Assignment

In Phase 4 of the targeting cycle, the commander reviews and approves the JIPTL. The targets on the approved JIPTL must now be assigned to the assets obtained from the capabilities analysis of the previous phase. The products of this phase are tasking orders that designate aircraft from specific bases to accomplish tasks at specific targets and times. In producing these tasking orders, it is important to allow for communication between the joint-level planners and their operational-level counterparts. Joint level planners use nominal data that might no longer be accurate in their planning process, and such interaction allows planners to rectify discrepancies that exist between assets or environmental conditions that were planned for, and those that are in existence. Because operational level planners might have access to more up-to-date information, they are capable of apprising joint planners of particular conditions or circumstances that might impact operations.

2.1.5 Phase 5 - Mission Planning and Force Execution

The level of detail in the tasking orders generated in Phase 4 is not sufficient to carry out operations with, and must be supplemented at the operational level. In Phase 5, the tasking orders of the previous phase are translated to detailed plans. In addition to these tasking orders, operational level planners have at their disposal the analytical reasoning used by the joint level planners in their assignment of particular assets to the targets, as well as the objectives and guidance issued from the commander.

Combat operations are dynamic, and planners must be capable of responding to changes in the enemy, environment, and allied assets. Because of this, operational planners, who are more connected with day to day combat operations, are often given liberty to make adjustments to the plans as they are being executed.

2.1.6 Phase 6 - Combat Assessment

Combat Assessment (CA) occurs after missions have been carried out, and helps planners determine if the effects on the enemy that were envisioned in the objectives, which are collectively referred to as the campaign plan, are being achieved through the operations. The product of the CA phase is a campaign assessment that is used by the JFC to shape his future decisions.

CA has three components: battle damage assessment (BDA), munitions effectiveness assessment (MEA), and future targeting or reattack recommendations (see Figure 2-2). BDA is a three-phased analysis of the damage inflicted on the targets after a mission is carried out on them. It encompasses the spectrum from the microlevel, where the individual target is examined, to the macrolevel, which examines the functionality of and effects on the comprising target system. The first phase examines the target itself, the second phase examines the functional consequences of the comprising target system, and the third phase projects the impact on the functioning of the system and behavior of the enemy.

To ascertain the effectiveness of the mission or campaign on a target system, it is necessary to first know the effect that was originally desired for the system. For this reason, BDA should be accompanied by a detailed understanding of the analysis performed during target development. In addition, establishing a standard criteria for assessing the degree to which a target has been successfuly disabled and the effectiveness of a campaign is important to maintaining consistency throughout the conflict.

Staff members of the MEA, the second component of CA, study the performance of the weapons used and the method in which they were applied. They compare expected performance with actual, and their findings are used to affect decisions

for the remainder of the conflict in the short-term, and future weapon choice and performance expectations in the long-term.

Future target nominations and reattack recommendations, as the last component of CA, tie both BDA and MEA feedback into the next iteration of the targeting cycle. With BDA, planners know whether the same targets or target systems must reappear on the JIPTL for further prosecution, and the results of MEA indicate whether any changes in the choice of weapon or method of employment should be made. This component closes the loop on the targeting cycle and allows decision-makers to learn from performance information that returns from the operational level.

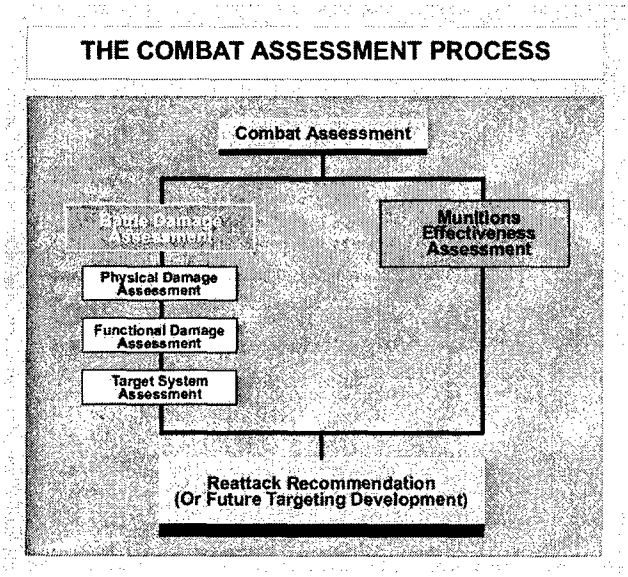


Figure 2-2: A representation of the Combat Assessment components, as presented by [20]

2.2 The Joint Air Operations Center

2.2.1 Operational Mapping of Functional Phases

As mentioned, the JTC is a functional depiction, in that it defines the required functions involved in performing targeting. These functions must now be assigned to different operational entities, and a framework defining the flow of the inputs and

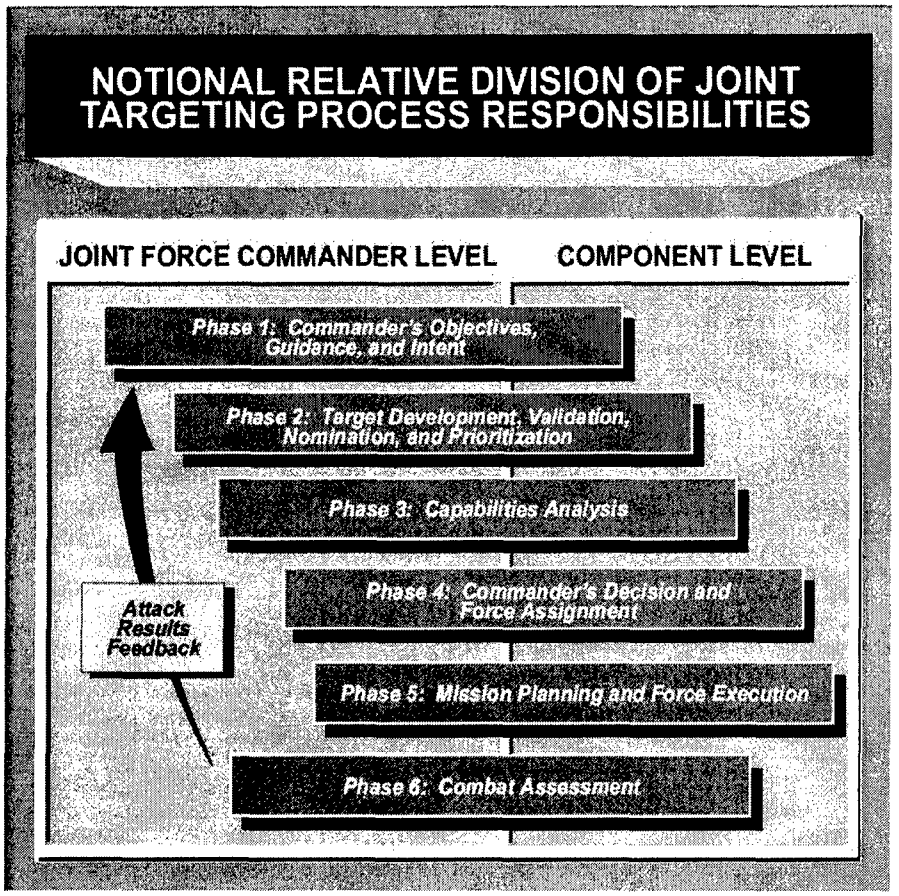


Figure 2-3: A functional to operational mapping of the targeting cycle, as presented by [20]

outputs among the operational entities must be established. Figure 2-3 provides a mapping of the JTC phases to the two main levels of players: the Joint Force Commander (JFC) Level, and the Component Level. As the figure shows, the JFC is responsible for the earlier phases where objectives and plans are developed, but assigns responsibility to the component commanders (i.e.,commanders of the air, land and sea force components) to execute plans in the later phases. In the last phase, results of the operations are fed back to the JFC, who evaluates them for the next iteration of the JTC.

Figure 2-3 reveals the fact that particular functions are often shared between the two defined levels of hierarchy. In practice, however, the distinction between the JFC and the component levels is even less clear with regard to the functions that they carry out. While the JFC has the ultimate authority to make decisions and guide efforts within his theater, he often requests a great deal of participation from the component commanders and the component command structure when making decisions with regard to their particular area of expertise.

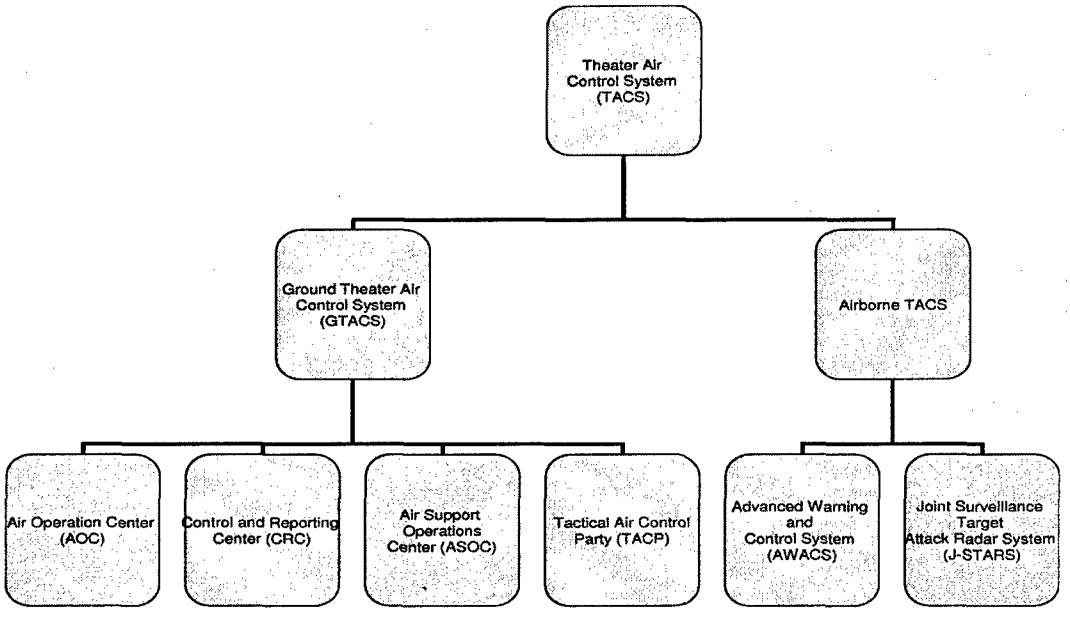


Figure 2-4: A hierarchical view of the Theater Air Control System

The air component commander is known as the Joint Forces Air Component

Commander (JFACC), and the command and control structure he controls is known as the Theater Air Control System (TACS, illustrated in Figure 2-4). The TACS is composed of both airborne and ground-based command and control (C2) elements. The Airborne elements include the Airborne Warning and Control System (AWACS) and the Joint Surveillance Target Attack Radar System (Joint STARS). Ground Theater Air Control System components include the Air Operations Center (AOC), Control and Reporting Center, Air Support Operations Center, and Tactical Air Control Party. The AOC, in turn is comprised of five divisions: Strategy; Combat Plans; Combat Operations; Intelligence, Surveillance and Reconnaissance (ISR); and Air Mobility. The AOC acts as the senior element of the TACS, and in this role publishes several documents that function to integrate the other components of the air C2: the Joint Air Operations Plan (JAOP), air operations directive (AOD), Air Defense Plan (ADP), Airspace Control Plan (ACP), Airspace Control Order (ACO), Air Tasking Order Special Instructions (ATO SPINS), Tactical Operations Data, and Operations Task Link. Because of the AOC's central role in the air targeting cycle, we now continue with a description of its five divisions.

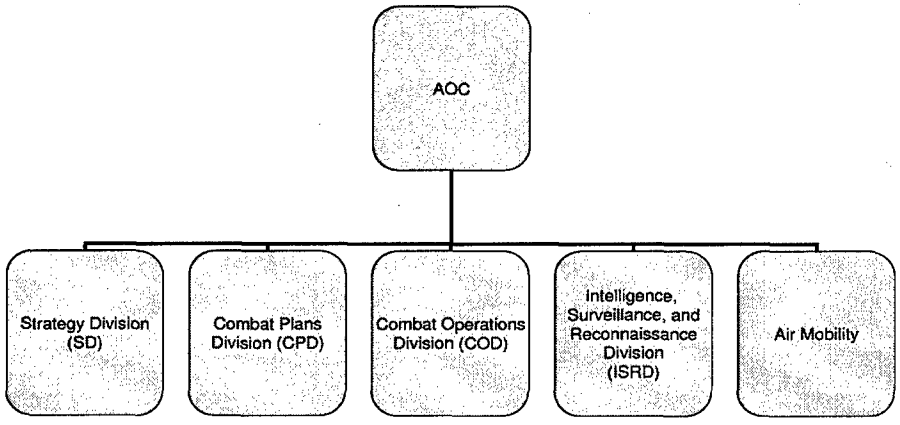


Figure 2-5: The Air Operations Center with its Divisions

2.2.2 The Strategy Division

The task of the Strategy Division (SD) is to develop, refine, disseminate, and assess JFACC air and space strategy. This strategy is presented in the form of a comprehensive JAOP, as a longer-term operational plan for the air component. The SD also initiates the shorter term air tasking cycle by publishing the AOD. In addition to these roles, the SD assesses the effectiveness and efficiency of air, space, and informational operations. Process inputs include higher headquarters policy, guidance and intent, and rules of engagement (ROEs) that come from the NCA, JFC, or JFACC. To perform its role in assessing operations, the SD also takes combat assessment summaries such as BDA and MEA reports. Process outputs include the JAOP, AOD, ATO SPINS, JFACC apportionment recommendation for the JFC, and operational assessment summaries [4].

2.2.3 The Combat Plans Division

The Combat Plans Division (CPD) is responsible for developing detailed execution plans for air and space operations. The plans assign particular air and space assets to perform tasks set forth by the JFACC to accomplish JFC goals. The CPD uses the JAOP, AOD, other JFACC guidance, the Joint Target List (JTL), the No-strike List, the Restricted Target List, daily component Target Nomination Lists (TNLs), component direct support sorties and common-use allocations, component air support requests, requests for airspace control measures, joint ISR collection requirements and associated data, inputs from AOC points of contact for specific ATO SPINS, Reconnaissance, Surveillance, and Target Acquisition Annex, and feedback from AOC divisions as inputs in the process. The major outputs of the CPD include the daily air apportionent recommendation, JIPTL, Master Air Attack Plan (MAAP), ATO with SPINS, ACO, ADP, ACP, Tactical Operations Data, Operations Task Link, C2 Communications Plan, C2 Architecture Plan, Close Air Support Plan, and ISR Synchronization Matrix [4].

2.2.4 The Combat Operations Division

The Combat Operations Division (COD) monitors and adjusts the execution of the current ATO. ATOs are adjusted when battlefield dynamics cause the true operating environment and conflict conditions to change from those that were planned for. One of the situations that might prompt adaptation of an already published ATO is the appearance of a previously unidentified Time Sensitive Target or Dynamic Target, that requires immediate attention in order to be successfully prosecuted. The COD responds to changes in the battlefield by command and control of air and missile defense operations, Information Operations, and by modifying the ATO through adding, removing, retargeting, or changing a sortie's mission (known as re-roling) [4].

2.2.5 The ISR Division

The ISR Division (ISRD) bears the responsibility for providing intelligence, and ISR operations to the JFACC and AOC. It is through information from the ISRD that targets are developed, validated, and nominated to the JIPTL. The ISRD drives the Air Tasking Cycle, as planning and execution of air and space operations rely on a threat and targeting picture for the theater. The assets that the ISRD directs are also responsible for measuring the effects and success of air and space operations. The ISRD conducts the planning for ISR missions, oversees their execution, and makes adjustments to plans already in execution when necessary. The ISRD also has a role in establishing Predictive Battlespace Awareness. Through the input that the ISRD provides, the JFACC is able to anticipate future battlespace conditions and respond quickly to emerging opportunities. Major process outputs of the ISRD include the ISR Synchronization Matrix; Reconnaissance, Surveillance, and Target Acquisition Annex; and the Intelligence Preparation of the Battlespace [4].

2.2.6 The Air Mobility Division

Though not a direct participant in combat planning and operations, the Air Mobility Division (AMD) is a vital component of the AOC. The AMD orchestrates intratheater airlift, intertheater airlift, and theater air refueling, ensuring that friendly forces receive the supplies and support necessary to perform their operations. Because the assets they schedule are global in nature, it is necessary for the AMD to coordinate with entities outside of the theater in addition to those within. The major outputs of the AMD are the airlift schedule, tanker schedule, and air mobility support schedule [4].

2.2.7 The Final Link - Operational Units

The final function that remains is carrying out the AOC-planned missions by the operational units. When the MAAP and ATO, published by the Combat Plans Division, make target-asset assignments, they specify only the origin and aircraft type of the assets to be used. It is the responsibility of planning cells that are imbedded within the units based at these locations to assign specific tail numbers and pilots for the missions. When making this final assignment, these planning cells are responsible for such matters as ensuring that pilots receive their required crew rest and only airworthy aircraft that have received necessary maintenance are scheduled. While at the bottom of the decision-making hierarchy, unit planning cells play an important role in the operations, as their decisions have direct impact on the availablility of their unit's assets. They are also responsible for sending accurate estimates for future aircraft availability to the higher planning levels. If their projections are inaccurate, future ATOs might assign assets that are not available or do not exist. These missions must then be filled by aircraft from other units, or the missions may need to be cancelled. In the case where the mission called for a coordinated effort with other air assets, such cancellations might cause big inefficiencies in operations, as the other assets involved might not be reassigned again within the current ATO.

Chapter 3

Model Development

As was depicted in the section on the Air Operations Center (Section 2.2), the decisions of the Targeting Cycle are delegated in a hierarchical manner among the different elements of the command chain. Namely, goals and strategy are developed at the National Command Authority level; theater targets are nominated and initial task assignments are made by the theater commander and AOC; and target-to-tailnumber assignments are made within the planning cells of the fighting units at the squadron level. The size of the problem and the many decisions associated with it make this decomposition necessary. Not only would solving the problem in its entirety be an insurmountable job for a single individual, but the problem actually lends itself to incremental solution and in the manner laid out by the current operational structure. Because this is already the practice operationally, we approach the Targeting Cycle by decomposing similarly. Furthermore, because of the time limitations of our research, we decide to limit the scope of our problem to a particular element of the decomposed problem. This chapter defines the scope of the problem that we address, and explains where our problem fits into the overarching targeting cycle. After establishing its position in the Targeting Cycle and in the AF hierarchy, we present our model for the problem and enumerate any assumptions that we make.

3.1 Scope

The focus of our research is guided by our estimation of the AF structure and its operations, if autonomous UAV units were to be realized. We envision that the UAV units would be dispersed in air bases throughout the battlefield, and that they would exist as fighting units capable of performing all the basic tasks in combat. The general structure of the AF would remain as it is, and the same decisions outlined in the Targeting Cycle would continue to exist, but we feel that many of these decisions would be automated.

For the purposes of this research, we generalize the phases of the Targeting Cycle [20] into three levels of decisions: high level, mid-level, and low-level decisions (Figure 3-1). The high-level decisions consist of the first four phases (Phases 1-3 and part of 4, which is shared with the mid-level) of the Targeting Cycle, and include the establishment of objectives and development of target lists to accomplish these objectives. The mid-level decisions correspond with phases 4 through 6, and include producing an ATO and assigning assets to the tasks listed on the ATO.

While the low-level is not specified in the Targeting Cycle, we mention it because of its pertinence in the development of autonomous UAV units. Low-level decisions include those of determining the paths aircraft will take to and from the targets, and the basic piloting decisions that must be made to ensure that the aircraft remain airborne and clear of obstacles. These decisions are typically managed at the squadron-level and by the pilots of the aircraft. While such decisions are assumed by the Targeting Cycle, they cannot be assumed with UAVs, and pose a significant hurdle in realizing autonomous UAV units. If the UAVs are to be fully autonomous, they must be capable of making such decisions on their own, and a "trajectory" planner must exist to guide the UAVs through their missions.

Aspects of the other levels of decisions can similarly be assumed by automated planners: a high-level planner could select targets for the JIPTL in a manner consistent with the objectives set forth for the conflict, and a mid-level planner could schedule and assign these targets to UAVs.

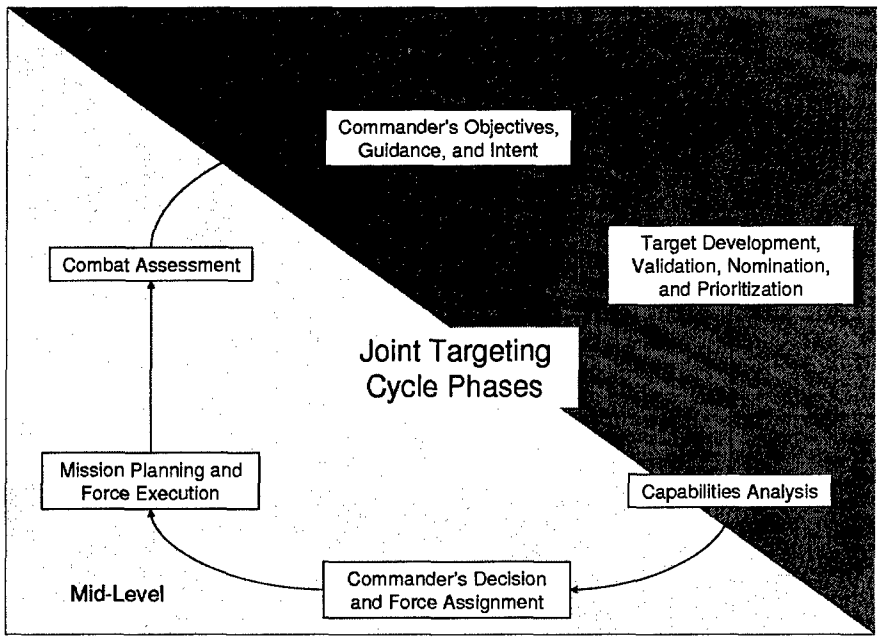


Figure 3-1: The subdivision of the Targeting Cycle by decision-levels

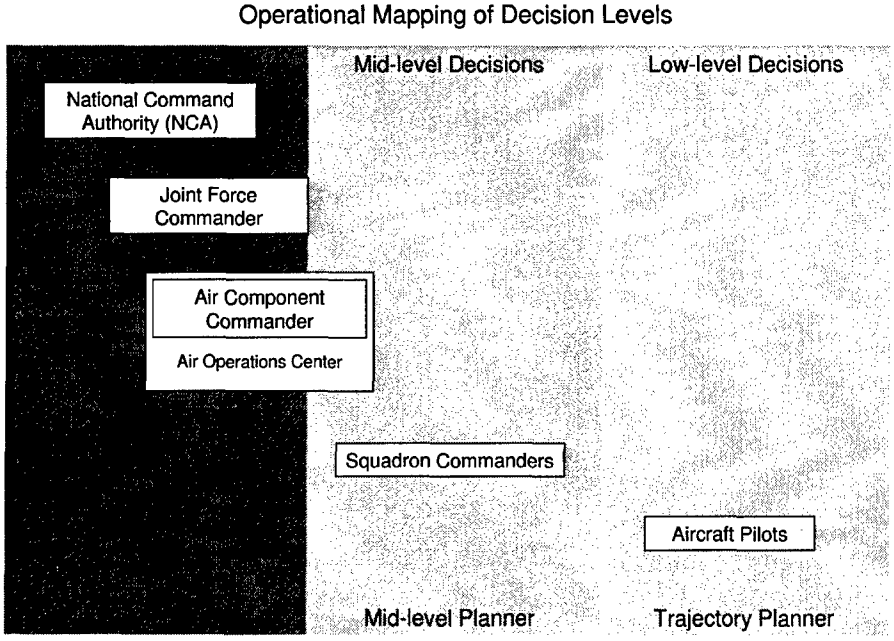


Figure 3-2: The operational mapping of the decision-levels

As illustrated by Figure 3-2, these decision-levels also correspond to the AF operational hierarchy. High-level decisions are those accomplished by the National Command Authority (NCA), Joint Force Commander (JFC), and Joint Air Component Commander (JFACC). Mid-level decisions are accomplished by the JFACC and his staff, as well as by the squadron planning cells. Low-level decisions are accomplished mainly by the pilots. The figure also depicts the planners that correspond to the three levels.

It is the mid-level problem of scheduling and assigning targets to UAVs to which we choose to confine the scope of our research. If current trends continue, we expect that the UAVs that comprise these units will have a considerable range advantage over their manned counterparts. The UAVs, then, will be capable of performing several tasks and visiting several targets in one mission. This fact makes mission planning for these UAV units a good candidate for optimization, as improved task allocation among the UAVs could mean more tasks accomplished by the units in each ATO cycle.

While we seek efficiency in the plans produced for these UAV units, we also want the plans be executable in practice. As mentioned in the introduction, a great deal of uncertainty exists on the battlefield, and we want robust plans that can be successfully executed even when reality on the battlefield differs from the expectations used in the planning. These two goals of optimality and robustness are often in conflict with one another, which is another reason the UAV Mission Planning Problem (UAVMPP) is an interesting research problem.

As indicated by Figure 3-2, the UAVMPP combines decisions made by both the JFACC and his AOC staff, and the squadron-level planning cells. This is due to the fact that we choose to join the decision of scheduling the tasks on the ATO with the decision of assigning the tasks to the UAVs. At present, the ATOs reach the squadron-level planning cells with target arrival-times already designated. The squadron-level planners are then only responsible for assigning the scheduled tasks to the pilots and aircraft in their squadrons. We feel that solving this assignment problem together with scheduling the arrival-times would present the opportunity for

improved efficiency in the resulting plans. While doing so adds to the complexity of the UAVMPP, our testing revealed that the model was still tractable. In addition, because our planner will be automated, the computational difficulty added by the joining of the two problems will not be borne by an individual.

3.2 The UAVMPP

This section further describes the UAVMPP and defines the inputs and outputs to the problem. It presents the assumptions that we make on the size and capabilities of the UAV units, and explains any other assumptions that we make in the model. As mentioned in the previous section, we envision that the UAV units would be a basic fighting force capable of performing the main combat operations. For simplicity, we only include what we believe to be the two most important combat operations into our model: Intelligence, Surveillance, and Reconnaissance (ISR); and strike. In our model, we define ISR as the gathering of further intelligence on an already established target, and can include searching for new developments or enemy movement at that target area. The strike of a target, in contrast, involves the firing upon and destruction of a specified target. We assume, that the targets, having already made it onto the JIPTL, have coordinates that specify their location in the combat theater. Due to the uncertainty in the battlefield, however, these coordinates might only specify the general location of the targets, as the targets might have moved since the intelligence on their location was gathered.

In keeping with present technology, we compose the UAV units with two types of UAVs: Type A UAVs, which are slower and have less range, but are capable of both ISR and strike functions; and Type B UAVs, which are faster, have greater range, but are only capable of ISR. Table 3.2 presents a comparison of the two UAV types. As it reveals, Type A UAVs are capable of a maximum of two strikes per mission. This is also motivated by current technology, as the smaller size of the UAVs presently limits them in the arsenal they can carry. We model each UAV unit with three Type A UAVs and one Type B UAV. This size was selected partially as a result of our vision

UAV Type	Type A	Type B
ISR capable	Yes	Yes
Strike capable	Yes, max of 2	No
Speed	Slower	Faster
Longevity	Shorter	Longer

Table 3.1: Comparison of the two UAV types

for the units, but also in large part because of the tractability issues associated with solving the UAVMPP for larger-sized units.

3.2.1 Inputs

The data inputs, which we use to define a particular scenario and which are required by the model, are the inter-target travel times, task service times, task accomplishment rewards, target opportunity window bounds, and UAV ranges. Inter-target travel times are approximated using straight-line distances between the target coordinates on a Cartesian plane. The travel times are then adjusted for by our assumptions regarding UAV speed. Namely, because of their speed advantage, Type B UAVs experience shorter travel-times than Type A UAVs. Inter-target distances, then, are specified by UAV type.

Service times are the times required by a UAV to accomplish a particular task at a target. The times are specified by target, task, and UAV type. We assume that strikes require more time to accomplish than ISR, and that, for a given target, Type B UAVs are able to accomplish ISR more quickly than Type A UAVs.

Successfully accomplished tasks have rewards assigned to them based on their significance in achieving the objectives outlined for the conflict. The rewards are specified based on target and task type.

We assume that targets may have opportunity windows, which specify the window of time during which the target is vulnerable to prosecution (either by strike, or by ISR). In addition to these windows, we solve the UAVMPP for an operating horizon that is comparable to the length of one ATO cycle. UAVs begin at ther home air base at the start of this operating horizon, and must return to the base by the close

of the horizon.

Finally, range values, which specify how long each UAV type may stay airborne, are also required as inputs to the model. The data inputs we mention are mean values, which are subject to the uncertainty that we define further in the next chapter. Just as the UAV types themselves are fabricated, the data inputs that correspond to them are also representative of what we believe to be logical values. We summarize these input items below.

Model Inputs

1. Inter-target travel times (by UAV type)
2. Task service times (by UAV type and target type)
3. Task accomplishment rewards (by target)
4. Target opportunity window bounds (by target)
5. UAV ranges (by UAV types)

3.2.2 Outputs

The output from the UAVMPP is a plan, similar to an ATO, which schedules tasks at the targets and assigns UAVs to each task. The plans specify the routes that the UAVs take through their assigned tasks, as well as the start times of each task. The routings are defined by "legs" that correspond to the inter-target paths selected by the planner on which the UAVs travel, and must begin and end at the air base. We enumerate the outputs of this model below.

Model Outputs

1. Task start times (indicates the UAV, target, and task to be performed)
2. inter-target legs used in plan (for each UAV)

3.2.3 Assumptions

In addition to the assumptions we have already stated, we also assume that UAVs are not lost, and that while a maximum of one ISR and one strike may be carried out at each target, it is not necessary that all possible tasks be accomplished at the targets. Our assumption concerning no UAVs being lost is made in order to simplify decision variables in the model. While our planner does not account for lost UAVs, we do capture some of the consequences of this possibility in our testing of the plans. In particular, UAVs lost because of enemy fire are factored into the model via mission reward uncertainty, and UAVs that are lost due to running out of fuel are considered lost by our testing and are not capable of visiting more targets.

The assumption made concerning one strike and one ISR per target is also an attempt at simplifying the model. We reason that subsequent strikes or ISRs at a target should not yield the same expected reward, and implementing a varied reward structure would make the model too complex. Making the tasks optional gives the planner more flexibility, and we feel this allows for improved plan efficiency. The planner will only choose the most profitable tasks in each planning horizon, and will not choose to accomplish a task if it necessitates the expenditure of more resources than it is worth. The noteworthy assumptions made by this model are summarized below.

Model Assumptions

1. UAV units contain 3 Type A UAVs and 1 Type B.
2. Only strike and ISR functions exist.
3. Type A UAVs are limited to 2 strikes, Type B are not strike-capable.
4. The general locations of targets are known apriori.
5. Scenarios are defined by inter-target travel times, target service times, target opportunity windows, target rewards, and UAV ranges.
6. Inter-target distances are approximated by straight-lines on the Cartesian plane.

7. All UAV routes must begin and end at the air base.

8. No UAVs are lost.

9. A maximum of 1 strike and 1 ISR at each target.

10. Tasks are optional.

3.3 Summary

In this chapter, we define the scope of the research to be the problem of scheduling the targets on the ATOs and assigning them to UAVs. This corresponds to the Targeting Cycle phase of *Mission Planning and Force Execution*, but also includes elements of *Commander's Decision and Force Assignment.* In terms of the operational framework, these decisions are made by both the AOC and the squadron-level planning cells. Our considerations for the UAVMPP scope include AF needs, problem richness, tractability issues, and improving efficiency and robustness. After establishing the scope of our UAVMPP research problem, we describe the UAVMPP in greater detail and explain the inputs, outputs, and assumptions made in the model. We develop the formulation that we use for the UAVMPP in the next chapter.

Chapter 4

Problem Formulation

In this chapter, we present a mathematical formulation to the UAV Mission Planning Problem (UAVMPP). A formulation helps us to better understand the real problem, and provides us with a means to solve it. Two countervailing interests exist when developing a formulation representative of a problem: the need to accurately model the problem, and the requirement that the model be computationally tractable. Our goal is a formulation that is rich enough to incorporate the elements of the problem of interest to us, but remains computationally tractable with the software and hardware available to us. In particular, we are interested in a tractable formulation that handles the elements of time, multiple vehicle functions and specifications, and is robust to the uncertainty that is inherent in the UAVMPP.

As a background, we briefly introduce linear programming, which forms the basis of the math formulation that we use for the UAVMPP. While the formulation we use is linear, it is special in that many decision variables are binary (accepting a value of either 0 or 1), and is thus referred to as a Mixed Integer Programming (MIP) problem. In particular, we present the problem as a Vehicle Routing Problem with Time Windows (VRPTW).

The MIP distinction is significant because it impacts the method of solving the problem. Furthermore, the formulation we use to represent the problem is dependent on the method we choose to solve the problem. A particular problem may be formulated in several ways, depending on the method chosen for solving it. Because of the

dependence of formulation and solution method, we present the two together in this chapter. We include the positive and negative aspects of the methods and formulations we present, and pay particular attention to whether and how they account for the uncertainty in the problem. The chapter includes a literature review, which illustrates the application of the methods we present to the UAVMPP or problems similar to it. We follow with our rationale for choosing to formulate the problem as a Vehicle Routing Problem with Time Windows that incorporates the robust optimization method suggested by Bertsimas and Sim [12], and is solved with a standard Branch-and-Cut algorithm. The chapter concludes with our formulation of the UAVMPP.

4.1 Linear Programming

The UAVMPP we present in Chapter 3 can be viewed as a resource allocation problem: a problem where a finite amount of resources must be allocated to a number of activities. Accomplishing these activities requires a certain amount of these resources, and the goal in resource allocation problems is to maximize the benefit that we receive from performing the activities, minimize the resources that are expended, or some combination of the two. For example, the UAVs at our disposal can be viewed as resources that must be allocated among the tasks at the target locations. We are interested in allocating the UAVs to optimize the reward that we gain from visiting the targets. Resource allocation problems are ubiquitous, and a field of research dedicated to solving them has developed in the past 60 or so years. This field, known as Linear Programming (LP), emerged as one of the most important scientific advances of the mid-20th century, continues to grow as new methods to handle new problems emerge, and continues to play an important role in many industries today.

Linear programming uses mathematical models to represent problems. It is linear in that the expressions that make up these models must be linear. It is considered to be "programming" not in the sense that it requires programming with a computer to accomplish, but in that it interests itself in planning. We, for instance, are interested in planning missions for the UAVs in our units. The appearance of programming

(planning) in its name also alludes to the fact that the output of an LP is a set of decisions that may be acted upon. These decisions are captured in the form of variables that set the level of each of the possible activities. An assumption of the general form of LP is that these decision variables are real numbers that can take fractional values (the assumption of divisibility).

The expressions that compose the math models used by LP typically consist of one objective function and a set of constraints. The goal of linear programming is to either maximize or minimize the objective function value (depending on the particular problem), which represents the rewards or costs (respectively) associated with the activities in the problem. The constraints may represent problem-specific conditions, such as the resources that a particular activity consumes, or may be general constraints that require that our decision variables be nonnegative.

The set of constraints specifies a feasible region in which all valid solutions (decision variable values) exist, as depicted in Figure 4-1 In two-space, bounds of the feasible region are lines, and in three-space they are planes. The task in LP, then, is to find the feasible solution that is optimal; that is, the feasible solution that provides the optimal objective function value. Today, this task can be accomplished for very large problems by using computers. One of the major discoveries of LP is that, as long as a feasible solution exists, an optimal one exists, and such a solution can be found at one of the corner points, which form at the intersection of two or more bounds. This is critical to the functionality of the simplex method, a novel procedure for solving LPs that was developed by George Dantzig in 1947. The fact that versions of the simplex method are still used routinely in today's solver packages speaks to the power of the method. Since the development of simplex, however, other methods such as interior point methods were also developed, and current software packages have utilized combinations of these methods to solve very large problems to optimality quickly.

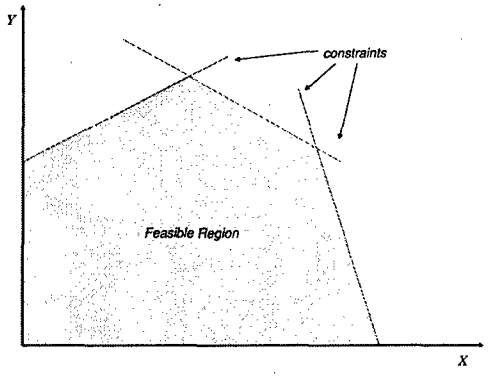


Figure 4-1: a graphical representation of a linear programming problem's feasible region

4.2 Integer Programming

The class of LP formulations we described are required to abide by the assumption of divisibility; decision variables were continuous and came from the domain of non-negative real numbers. This, however, is not the case for many problems that we face today. As is the case with many "real world" problems, our UAVMPP has decision variables that are constrained to be discrete. More specifically, our decisions for whether to send a UAV to a target or not are binary, in that only two possible outcomes exist: yes (1) and no (0). Formulations like ours, which include both continuous and discreet decision variables, are known as Mixed Integer Programming (MIP) problems, and fall under the domain of Integer Programming (IP).

This distinction is a simple change in terms of formulation (and is done by adding the constraint: $x \in int$ or $x \in bin$ for each x that is an integer or binary variable), but has more profound implications with regard to the methods for solving the problem. However, before discussing methodology further, we will continue with our discussion on formulation. More specifically, we introduce the class of MIP problems that our UAVMPP belongs to: Vehicle Routing Problems (VRP) [27].

4.2.1 The Vehicle Routing Problem

When formulated as an IP, our UAVMPP fits under the more general class of problems known as Vehicle Routing Problems (VRP). VRPs use a network of nodes and arcs to represent the problem. One node is used to represent a supply node, or depot, at which a group of vehicles is based. These vehicles transport a specified commodity from the depot to the other demand nodes, referred to as "customers" in what follows, which represent locations that possess a demand for the given commodity. All travel is modeled along a series of arcs that connect the nodes. The depot and customers might have a specified supply or demand (respectively) of the commodity, and the arcs might have an associated cost that represents the cost of travel along them. Vehicles might be limited in the amount of the commodity they are capable of carrying, and might be limited in the distance along the arcs that they are capable of travelling before they must return to the supply node.

In the general-form VRP, demand at all demand nodes must be satisfied, and all vehicles must return to the depot at the end of their tour. Because only one vehicle is allowed to make a delivery at a particular node, this vehicle must be capable of satisfying all of the demand at that node. The VRP, first introduced by Dantzig [41], is a rich class of problems, and is flexible to fit many circumstances. The constraint mandating that demand at all customers be met can be relaxed, vehicle range and capacity can be unlimited, customers can have multiple vehicles making deliveries, and time can be considered in the problem. In the case where time is considered, the time required to make a delivery can vary by customer, and vehicles can be given an opportunity window during which the delivery to a particular customer must be made. Decision variables specify whether a particular vehicle makes a delivery to a particular customer. In the case where time is considered, an additional decision variable specifies when the delivery is to be made.

The flexibility of the problem highlights its broad applicability in different industries and sectors of society. A VRP, for instance, can describe the routing of delivery trucks from a supply depot to retail outlets. It can also be used to characterize postal

service air operations, with the depot being a hub airport, customers being other airports in the network, and vehicles being cargo planes delivering packages. In the case of the UAVMPP, the depot represents the air base where all of the UAVs (the vehicles in our formulation) are stationed. The customers are targets that we are interested in performing tasks at, and the demand at these targets can consist of strike or ISR missions. UAVs have range limitations, and capacity limitations can represent the limited number of ordnances they are able to carry for strike missions.

The objective function of a VRP might call for the minimization of the cost incurred by traversing the arcs, might call for the maximization of the benefit received from making deliveries to the customers, or some combination of the two. While some constraints might be problem specific, VRPs also have common constraints that ensure that vehicles perform continuous tours that start and end at the depot. We present a formulation of the VRP where decision variables not only dictate if a particular vehicle will make a delivery for a customer, but also when such a delivery will be made. With this element of time included, the "subtour elimination constraints", as they are called, become unnecessary, as vehicle tours are guaranteed to be continuous. We now present a representative VRPTW (Vehicle Routing Problem with Time Windows) problem. Table 4.1 provides a description of the elements of this formulation.

Sets	
N	The set of nodes. It is composed of $\{0\}$, the depot, $n+1$, the return depot, and the customers
A	The set of arcs.
K	The set of vehicles.
$\Delta^+(i)$	the set of arcs leaving node i
$\Delta^-(i)$	the set of arcs entering node i
Decision Variables	
x_{ijk}	binary, equals 1 when vehicle k traverses arc between nodes i and j
w_{ik}	nonnegative real number, represents the arrival time for vehicle k at node i
Constants	
c_{ij}	the cost for traversing ark (i,j)
s_i	the time required to service customer i
t_{ij}	the time required to traverse arc (i,j)
a_i,b_i	the earliest and latest times that node i can be serviced, respectively.
e,l	the absolute earliest and latest times, which form the time horizon of the problem.
d_i	the demand at customer i
c	the carrying capacity of each vehicle

Table 4.1: Elements of the VRPTW

$$\min \quad \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ijk} \tag{4.1}$$

$$\text{s.t} \sum_{j\in\Delta^+(0)} x_{0jk} = 1 \quad \forall k \in K \tag{4.2}$$

$$\sum_{i\in\Delta^-(j)} x_{ijk} - \sum_{i\in\Delta^+(j)} x_{jik} = 0 \quad \forall k \in K,\ j \in N \tag{4.3}$$

$$\sum_{i\in\Delta^-(n+1)} x_{i,n+1,k} = 1 \quad \forall k \in K \tag{4.4}$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \quad \forall k \in K,\ (i,j) \in A \tag{4.5}$$

$$a_i \sum_{j\in\Delta^+(i)} x_{ijk} \leq w_{ik} \leq b_i \sum_{j\in\Delta^+(i)} x_{ijk} \quad \forall k \in K,\ i \in N \tag{4.6}$$

$$e \leq w_{ik} \leq l \quad \forall k \in K,\ i \in \{0, n+1\} \tag{4.7}$$

$$\sum_{i\in N} d_i \sum_{j\in\Delta^+(i)} x_{ijk} \leq c \quad \forall k \in K \tag{4.8}$$

$$x_{ijk} \in \{0,1\} \quad \forall k \in K,\ (i,j) \in A \tag{4.9}$$

(4.1) represents the objective function to minimize travel costs. Constraints (4.2) and (4.4) require that all vehicles leave and return to the depot, respectively. The depot is represented as two nodes: a departure node and an arrival node. Constraints (4.3) are flow conservation constraints that require a vehicle to leave a node if it visits it. Constraints (4.5) ensure adequate separation between contiguous arrival times. Arrival times at sequentially visited nodes must allow for the time required to serve the first node and travel to the second node. (4.6) require that arrival times occur within the time windows of their respective targets. (4.7) ensure that all vehicles leave and return to the depot within the time horizon specified by the problem, [E,L]. (4.8) ensure that vehicles have enough capacity to fully meet the demand of the customers they serve, and (4.9) require the x_{ijk} decision variables to be binary. The w_{ik} are non-negative as long as e and l are.

4.2.2 Methodology

As mentioned in Section 4.2, the MIP distinction our UAVMPP holds, carries with it implications in terms of the method we use to solve the problem. MIP problems

cannot be solved by using the simplex method alone. LP relaxation solutions (where integer constraints are relaxed) are not guaranteed to be integer, and finding an optimal integer or binary solution from the LP relaxation is a nontrivial task. Simply rounding to the nearest whole number will not necessarily result in the optimal integer solution, and this solution might in fact be severely suboptimal or infeasible.

Several methods have been developed for tackling this problem of finding the optimal solution to IP and MIP problems. These methods are classified as either exact methods, approximations, or heuristics. Exact methods provide the optimal IP solution, but may take an exponential number of iterations; approximations provide suboptimal solutions in polynomial time, but include a bound on the degree of suboptimality; and heuristics likewise provide suboptimal solutions in polynomial time, but with no bound. The methods we review in this section are the exact methods of Dynamic Programming, Branch-and-Bound / Branch-and-Cut algorithms, and the heuristic methods of Local Search methods and Simulated Annealing. With each method, we attempt to give a review of research that has applied the corresponding method to problems similar to ours. Namely, we review literature in which the method in question was applied to a VRP, or, when it is available, literature applying the method to the UAVMPP.

Dynamic Programming

Dynamic Programming (DP) decomposes a problem into a sequence of stages, where decisions must be made at each stage. The problem is defined by "states" that define the state of the system and encapsulate all relevant previous decisions. Because of this, the system state in a subsequent stage is dependent only on the current state and decision. Optimal decisions for future stages, then, are not dependent on previous decisions, but only on the current state. This configuration can be depicted as a network, as illustrated by Figure 4.2.2. States of consecutive stages are linked together via arcs that represent decisions at the corresponding stage. The arcs have associated transition costs, and arcs are present only for feasible state transitions. No arcs, for example, can connect to states in the same or previous stages. All possible states of the

problem are enumerated, and the problem is solved through backward induction: the states corresponding to the last stage of the problem are solved to optimality, followed by the states of a previous stage and so on. The problem formulation is not the typical LP formulation with objective function and constraints, but consists of a recursive objective function of the form $C(S,k) = \min_{m \in S \setminus \{k\}}(C(S \setminus \{k\}, m) + c_{mk}), k \in S$, where $C(S,k)$ is the cost of the optimal solution at node k, S is the set of all nodes in the stages already solved for (the state here is defined by (S,k)), and c_{mk} is the transition cost from node m to node k. This objective function is solved until S includes all possible nodes.

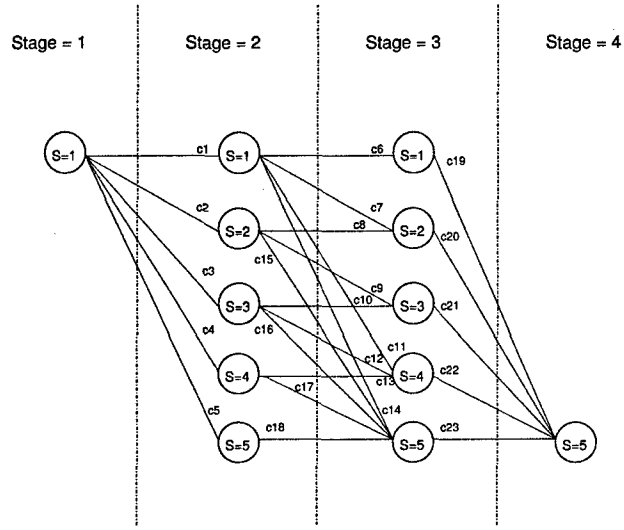


Figure 4-2: A network representation of a dynamic programming problem

Not only does Dynamic Programming provide an optimal solution to the problem from the initial stage and state, but it provides optimal decisions for to every state in the future stages as well. If at some stage, a user finds himself at a suboptimal state, he is able to know the optimal course of action from that state without solving the problem again from the new state.

The Dynamic Programming reformulation and solution algorithm method can also be used to model and solve stochastic problems. In such problems, transitions to future states are dependent not only on the current state and decision, but also on a

probability function at that stage and state. Once a decision is made at a particular state, the decision is subjected to a probability function in order to determine the future state. Figure 4-3 provides a network representation of the process. It fits well into a problem such as the UAVMPP, which is stochastic. The optimal course of action is perfectly outlined regardless of future state realizations, and the plan can be updated with little or no additional computational costs as realizations are manifested.

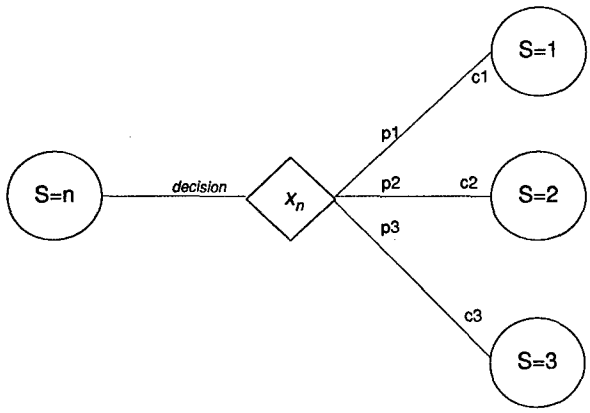


Figure 4-3: an example implementation of stochastic Dynamic Programming

Zarybnisky [43] uses a DP formulation known as Partially Observable Markov Decision Process (POMDP) to solve the targeting cycle problem. The targeting cycle problem modeled by Zarybnisky takes a higher-level approach to the resource allocation, as well as a broader scope, in that it concerns itself with target acquisition as well. Knowledge of the present state is stochastic in POMDP, and the user does not know with certainty the state that he is in. Specifically, the user is aware of his own resources, but not of those of the enemy. The user bases his decisions not necessarily on the true condition of the system, but rather on the information state, I_k, that comprises all of the actions he has performed and observations of the system that he has made. Instead of considering a state space, the method relies on a belief state, which is a probabilistic mapping of the state space that accounts for the uncertainty with regard to the enemy. POMDPs were proposed by Drake [24], but

Zarybnisky adapts a hybrid method, developed by Yost [42], that combines an LP resource allocation master problem with several POMDP subproblems that generate improved contingency plans at each stage [42]. The master problem is solved using a standard MIP method, while the POMDPs are solved using the Linear Support and Incremental Pruning algorithms.

While Zarybnisky's POMDP model considers the uncertainty in and limited knowledge of the enemy's condition, it does not consider uncertain the condition of friendly assets, nor the amount of resources it will require to accomplish particular tasks on enemy objects. He takes a high level approach on the resource allocation, and measures the assets required by tasks in terms of numbers of planes or weapons. As mentioned earlier, we desire a higher level of fidelity for the UAVMPP, that better utilizes the longer operating horizons of UAVs by allowing them to accomplish multiple tasks in one of Zarybnisky's time-steps. Others have also utilized Dynamic Programming to solve similar targeting or VRP problems: [9] [2] [10]

The main criticism of DP is its tractability. DP suffers from what is known as the *curse of dimensionality.* The algorithm calls for enumerating all possible states and solving each to optimality. The number of states in a problem increases exponentially as the number of variables (dimensions) increases, and the number of possible states is equal to all possible combinations of all possible values of all variables. The state space "blows up" as the number of variables increases, and DP can only be used to solve small problems.[13]

Branch-and-Bound & Branch-and-Cut

Branch-and-Bound differs from DP in that it retains the standard IP formulation. In our case, then, we solve the UAVMPP using a VRPTW formulation. Branch-and-bound uses a "Divide and Conquer" approach to intelligently search through feasible integer solutions. The algorithm divides the feasible region into subregions, that can then be solved separately. Because the difficulty of finding optimal integer solutions in subsections is often similar to that of solving over the initial feasible region, the algorithm solves for lower bounds to these integer solutions instead (for

minimization problems; it solves for upper bounds in maximization problems). These lower bounds, which typically are the LP relaxations to the problem, are much easier to solve. The method also calls for occasionally finding candidate integer solutions in the subregions to serve as upper bounds. These upper bounds allow the algorithm to intelligently select which regions to pursue further, as regions for which the lower bound (LP relaxation) is greater than the current upper bound can be weeded out (or "pruned").

Sample Branch-and-Bound Algorithm

1. Select an active subproblem.
2. If the subproblem is infeasible, delete it; otherwise, compute its lower bound.
3. If lower bound is greater or equal to current upper bound, delete the subproblem.
4. If lower bound is less than current upper bound, solve the optimal solution or divide the subproblem into further subproblems.

Figure 4-4: An example of a Branch-and-Bound algorithm, adapted from [13]

The Branch-and-Cut algorithm differs from that of Branch-and-Bound in that it applies "cuts" to the subproblems before solving lower bounds for them. Cuts are further constraints that are added which reduce the feasible region of the subproblem without eliminating any candidate integer solutions within the feasible region. The cuts are added to speed the process of finding an optimal integer solution. While cuts can be very effective in reducing solve-times, cuts are problem specific, good cut constraints are not easy to come by, and adding excessive cuts can even lengthen solve-time, as the added constraints will cause bounds to take longer to compute.

Because Branch-and-Bound and Branch-and-Cut methods are very computationally intensive, they are often impractical for large scale problems. Lau and Kumar address this issue in their research by performing Branch-and-Bound in parallel on a Network of Workstations (NOW) [29]. Bard, et al. [8] solve a VRPTW using their

modified Branch-and-Cut algorithm [8]. The algorithm finds continually better lower bounds by solving a series of relaxations at each step that add newly found valid constraints. The upper bound for the algorithm is updated using a Greedy Randomized Adaptive Search Procedure (GRASP). In order to mitigate the additional computational requirement brought in by using cuts, their algorithm calls for solving a separation subproblem to find cuts in their cut pool that are violated. Other works that investigate the use of Branch-and-Bound or Cutting-Plane algorithms for VRP problems are [14] [21] [22] [25] [5] [38]. Reference [3] provides a good overview of current practice on solving the Weapon Target Assignment Problem; the reference presents Branch-and-Bound algorithms but also investigates heuristics.

Local Search

Local Search, like the other heuristics we discuss, came about as an attempt to circumvent the tractability issues confronted by the exact methods. They usually provide no bound on the optimality of their solutions, but generally provide a good solution quickly. The general concept of the Local Search is to find and evaluate a solution within the feasible region, and then to evaluate neighbors to this solution. If a neighbor proves to yield a better objective function value, the algorithm moves to it and explores its neighbors. If not, the algorithm has found a local optimum, or a feasible solution that was proven to be at least as good as its neighbors. A typical Local Search algorithm will repeat this process several times with different initial starting solutions. What constitutes a "neighbor" to a solution can be interpreted differently, and the definition of a neighbor is often picked somewhat arbitrarily. Local Search is known as a heuristic because it is not guaranteed to find the global optimal solution, but only a local optimum. Popular Local Search algorithms include Ant Colony, Genetic Algorithms, Neighborhood Search, and Tabu Search algorithms. Reference [28] [1] [11] [6] [26] [30] [34] [35] [15] [37] [16] for further use of Local Search algorithms for the VRP. [36] contains lecture slides concerning the VRP and different solution methods to it.

Simulated Annealing

Simulated Annealing is a generic method that builds on the Local Search method. As mentioned, Local Search is only capable of identifying whether or not a solution is a local optimum; in general, no guarantees on global optimality can be established when using Local Search. Simulated Annealing modifies the Local Search algorithm by occasionally allowing for a move to a less optimal solution. While counterintuitive, such moves give the algorithm an opportunity to escape from local optima that are not the global optimum.

We now explain the behavior of the algorithm with a maximization problem. The algorithm starts with a feasible solution, x, and picks a neighboring solution, y, at random. If the benefit of solution y is greater than that of x ($b(y) \geq b(x)$, the algorithm moves to solution y. If $b(y) < b(x)$, it will move to solution y with probability $e^{-(b(x)-b(y))/T}$, where T is a constant known as the temperature. It has been shown that the algorithm arrives at the global optimum with a very high probability after being run for a very long time at a very low "temperature." The low temperature causes the algorithm to take a long time to escape local optima, and it is for this reason that the algorithm must be run for a long time. As this is not practical, a compromise is to vary temperature with time, beginning with a larger temperature that allows the algorithm to explore more of the feasible area, and subsequently decreasing the temperature as the algorithm progresses. When the temperature is varied in such a manner, it is known that the probability of the current solution being the global optimum equals 1 at the limit where time approaches infinity. Simulated Annealing is a general method, and while it has been shown to work very well for certain hard to solve problems, it often underperforms special purpose methods designed around those specific problems. Refer to [19] [7] [33] [40] for further use of Simulated Annealing for the VRP.

4.2.3 Discussion

Up to this point, all the methods for solving the UAVMPP that were presented have suffered from one or more of the following flaws: they were computationally intractable, they provided no guarantees on optimality, or they did not account for uncertainty in the problem. Branch-and-Bound methods are exact, but are computationally intensive and do not account for data uncertainty. Dynamic Programming is exact and does account for uncertainty, but suffers from the curse of dimensionality and is not practical for solving our problem because of its computational intensity. The heuristics we mentioned appear to provide reasonably good solutions, but generally do not account for uncertainty, and do not guarantee optimality (and often don't give bounds on their suboptimality either).

While Linear Programs and Integer Programs do not consider uncertainty in their base form, research has gone into modifying this base formulation so that it does account for uncertainty. The solution to the modified problem is "robust" in that it accounts for uncertainty in the data and remains feasible under many realizations of this data. In the following section, we introduce the field of Robust Optimization.

4.3 Robust Optimization

A great deal of effort has been directed at incorporating the stochastic nature of problems into the decision process. To this end, Robust Optimization research attempts to transform the current linear programming formulation so that it protects against a degree of uncertainty.

As mentioned in the Linear Programming Section 4.1, LP problems are typically formulated with an objective function and a set of constraints that define a feasible region in which we might find a solution. Robust Optimization protects against uncertainty by considering only a subset of the feasible region from the original problem. The rationale for doing this is intuitive. While optimal solutions for problems with linear objective functions will exist on the boundary of the feasible region (and a typical MIP solution method will consequently terminate with a solution on or near

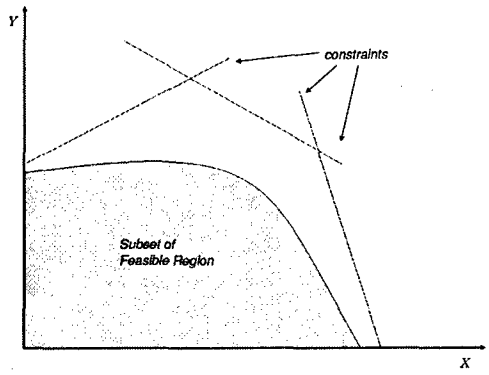


Figure 4-5: Robust Optimization considers only a subset of the feasible region

this boundary), so does the uncertainty, which exists within the coefficients of the boundary-defining constraints. Realizations of the uncertain coefficients might produce a realized feasible region that lies within the original one, causing solutions on the boundary of the original feasible to be infeasible. We can protect from a degree of uncertainty by only considering solutions that exist within a subset that is separated from the boundary. In our mission planning problem, for instance, we might be uncertain of the exact range of an aircraft, of the distances between the targets, and of the time required to service these targets. These values are captured in the coefficients of the constraints, which collectively define the boundary of the problem's feasible region. We can protect from the uncertainty in these values by selecting only from solutions that are contained within a subset of the feasible region (see Figure 4-5).

This field has attracted a fair amount of attention in recently, and several different methods for Robust Optimization have been developed. These methods differ in the way that they define the subset of the feasible region over which we optimize. The first Robust Optimization method we consider is that suggested by Soyster [39]. In his approach, Soyster considers a subset that is feasible to all data uncertainty. The result is a very robust solution, but one that gives up a large degree of optimality. Ben-tal and Nemirovski suggest another method, which approximates the true feasible area by an elliptical subset of the original feasible region. While the approach takes an interesting perspective on the feasible region, it adds complexity to the original

problem by introducing a constraint that is conic. In their paper, "The Price of Robustness" [12], Bertsimas and Sim suggest another method, which expands from the Soyster method and protects against a user-defined level of uncertainty. A general overview of the Bertsimas-Sim method is provided in the following section.

4.3.1 The Bertsimas-Sim formulation

As mentioned, the Bertsimas-Sim formulation builds on the Soyster method. It assumes that all coefficients have ranges in which they can exist. The coefficients used in the nominal case (the nominal case is our original problem, which disregards data uncertainty), are now considered mean values, and realizations of these coefficients can now assume values within ranges centered at these means. For example, if A_{ij} is a coefficient in the original problem, we define a halflength, $\hat{A}_{ij}$, and the allowable range for A_{ij} is then $[A_{ij} - \hat{A}_{ij}, A_{ij} + \hat{A}_{ij}]$. It is not necessary that all coefficients in a constraint are uncertain, and $\hat{A}_{ij}$ is defined only for all $j \in J_i$,where J_i is the set of uncertain coefficients in constraint i. $|J_i|$ is less than or equal to the number of coefficients in constraint i.

The Bertsimas-Sim formulation allows the user to specify protection levels. These values, denoted as Γ_i, dictate the number of coefficients that the formulation will protect from uncertainty in constraint i. Γ_i is only meaningful for values $0 \leq \Gamma_i \leq |J_i|$, but need not be integral. In the case where $|\Gamma_i|$ is not integral, $\lfloor \Gamma_i \rfloor$ coefficients and $\Gamma_i - \lfloor \Gamma_i \rfloor$ of another coefficient receive protection. The coefficients that receive protection are given their worst-case values. In other words, they are given a value that corresponds with one of the bound-values of their range. The formulation decides which of the uncertain coefficients will receive protection by maximizing $\sum_j \hat{A}_{ij} y_j$ when summed over all j coefficients selected to receive protection. Intuitively speaking, the formulation finds the Γ_i coefficients that together shrink the feasible region the most. This formulation is illustrated in the system of equations below.

$$\max \; c'x$$

$$
\begin{aligned}
\text{s.t.} \quad & \textstyle\sum_j A_{ij}x_j + \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{A}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{A}_{it_i} y_t \right\} \leq B_i \quad \forall i \\
& -y_j \leq x_j \leq y_j \quad \forall j \\
& L \leq x_j \leq U \quad \forall j \\
& y_j \geq 0. \quad \forall j
\end{aligned}
$$

The formulation retains its original structure but has the constraint $-y_j \leq x_j \leq y_j$, and the maximization subfunction added to it. The added constraint and the decision variable, $\mathbf{y}$, are required to ensure that protection is added (and not removed) even for negative values of $\mathbf{x}$. The maximization subproblem chooses the $|S_i|$ coefficient hat values, $\hat{A}_{ij}$, and the $|t_i|$ portion of a remaining hat value so that the function $\left\{ \sum_{j \in S_i} \hat{A}_{ij} y_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{A}_{it_i} y_t \right\}$ is maximized for constraint i. $|S_i| + |t_i| = \Gamma_i$, and t_i is required only when Γ_i is not integral.

This formulation is no longer linear because of the maximization subproblem. Bertsimas and Sim show that this formulation can be written in an equivalent manner that is linear. This transformation is achieved by taking advantage of strong duality. The dual of the maximization subproblem is taken and its variables are included among the constraints of the original problem. When the resulting problem is solved, feasibility is secured in the original problem and the dual problem, and optimality is achieved in the original. We give the transformed problem below (due to [12]).

$$
\begin{aligned}
\max \quad & C'x & & & (4.10) \\
\text{s.t.} \quad & \textstyle\sum_j A_{ij}x_j + z_i\Gamma_i + \sum_{j \in J_i} p_{ij} \leq B_i & \forall i & & (4.11) \\
& z_i + p_{ij} \geq \hat{A}_{ij} y_j & \forall i, j \in J_i & & (4.12) \\
& -y_j \leq x_j \leq y_j & \forall j & & (4.13) \\
& L_j \leq x_j \leq U_j & \forall j & & (4.14) \\
& p_{ij} \geq 0 & \forall i, j \in J_i & & (4.15) \\
& y_j \geq 0 & \forall j & & (4.16)
\end{aligned}
$$

$$z_i \geq 0 \qquad \forall i \qquad (4.17)$$

In the formulation, the objective, (4.10) is unchanged, constraints (4.11) are modified from the nominal (original) formulation, (4.14) are unchanged from the original formulation, and the remaining constraints are additions made by the Bertsimas-Sim formulation. Constraints (4.15) through (4.17) ensure nonnegativity for the "robust" variables: p, y, and z. The variables, p and z, are the dual variables. As mentioned earlier, the decision variable, y, and its corresponding constraints, (4.13), are necessary only when the original decision variable, x, is allowed to take negative values. In such cases, y, which is set to $|x|$ by constraints (4.13), ensures that the half-values are added to (and not subtracted from) constraints with the effect of always making the constraints more restrictive. The set of constraints represented by (4.12) force the sum of z_i and p_{ij} to $\hat{A}_{ij}$ for all i and j. The z_i and p_{ij} variables are then added to their corresponding constraint (4.11). They affect only basic variables (variables that are included in the optimal solution), and of those, only the Γ_i that are protected by the formulation. The Γ_i coefficients that are selected must come from J_i, the set of uncertain coefficients in constraint i. The following paragraph gives a detailed explanation of how the formulation decides which Γ_i coefficients to protect. It is not necessary for an understanding of our research, but is included as we feel that it provides the reader with a better appreciation of the Bertsimas-Sim formulation.

Because y_j multiplies $\hat{A}_{ij}$ in constraints (4.12), if y_j (and thus x_j) is zero in the optimal solution, the corresponding dual variables will likewise be set to zero (the LP will minimize the dual variables, and set them to zero if it can). In the case where $\hat{A}_{ij}y_j$ is nonzero, the values of p_{ij} and z_i are assigned in the following way: when Γ_i is equal to zero, z_i will assume the value of the largest $\hat{A}_{ij}y_j \quad \forall j \in J_i$ in constraint i. In this way, all p_{ij} can be set to zero, all of the half-values can "hide" behind the largest $\hat{A}_{ij}y_j$, and nothing is added to the nominal constraint i, as z_i is multiplied by a Γ of zero in it. In the same way, as Γ increases from zero, z_i assumes the value of the Γth largest $\hat{A}_{ij}y_j$. This allows all of the $\lfloor\Gamma\rfloor + 1$ through $|J_i|$ $\hat{A}_{ij}y_j$ to "hide" behind z_i. For these $\hat{A}_{ij}y_j$, p_{ij} are zero, and for the $\hat{A}_{ij}y_j$ that are larger than z_i, the

p_{ij} assume the remainder of $\hat{A}_{ij}y_j - z_i$. Any larger of a value for z_i, and we penalize the constraint more than necessary, because z_i is multiplied by Γ. In fact, assigning z_i in the way described will add the least amount to the Left Hand Side (LHS) of our original constraint, which is favorable. Put another way, assigning any other value to z_i will cause more to be added to our original constraint and will make it more restrictive. As noted, only the Γ largest $\hat{A}_{ij}y_j$ are added to the LHS of constraint i. It is not simply the magnitude of the half-values that is considered in selecting which coefficients are protected, but the magnitude of the half-values multiplied by the decision variables in the optimal solution. A coefficient with less uncertainty but that multiplies a decision variable that is heavily utilized in the optimal solution, might then be protected instead of a highly uncertain coefficient that is multiplied by a less-utilized decision variable. In this sense, the Bertsimas-Sim formulation also considers the objective function in its selection of protection, as the utilization of decision variables is based on their value to the objective function.

4.4 UAV Mission Planning Problem Formulation

We now discuss our choice of formulation and methodology for the UAVMPP: a VRPTW modified to incorporate the Bertsimas-Sim formulation and solved using a commercially-available Branch-and-Bound solver. We utilize a VRP with time windows as this seemed a natural fit for the UAVMPP. The time windows allow us to specify "opportunity windows" during which targets are available, and the time of arrival now becomes a decision variable. As mentioned earlier, another attractive feature of the VRPTW is that the tedious subtour elimination constraints are no longer necessary, as the inclusion of the time variable ensures that no subtours will occur. To aid the reader's understanding of our formulation, we present it incrementally. Below is the UAVMPP as a VRPTW, before the Bertsimas-Sim formulation is incorporated into it. Table 4.2 defines the sets, decision variables, and constants found within the formulation. Table 4.3 gives a line by line description of the formulation. It divides the formulation into sections: the objective function; structural constraints, which

are deterministic constraints that ensure logical solutions; constraints that contain stochastic coefficients and are subject to uncertainty; and binary and nonnegativity constraints.

Sets	
N	The set of nodes. It is composed of $\{0\}$, the depot; T, the set of targets; and n, the return depot ($N = T \cup \{0, n\}$)
K	The set of UAVs. It is composed of the unions of the set of Type A UAVs, denoted UAV_A and the set of Type B UAVs, denoted UAV_B.
Decision Variables	
$ISR_{k,i}$	binary, equals 1 when ISR is performed at node i with UAV k.
$STR_{k,i}$	binary, equals 1 when a strike is performed at node i with UAV k.
$x_{k,i,j}$	binary, equals 1 when UAV k traverses arc between nodes i and j.
$w_{t,k,i}$	real number, represents the time UAV k performs task t at node i $t \in \{strike, ISR\}$.
Constants	
M	a large value.
$C_{t,i}$	the reward for accomplishing task t at target i. $t \in \{strike, ISR\}$.
A_i,B_i	the earliest and latest times that node i can be serviced, respectively.
$S^1_{m,i}$	The required service time for Type A UAVs to accomplish task t at node i. $t \in \{strike, ISR\}$.
S^2_i	The required service time for Type B UAVs to accomplish ISR at node i.
$D_{u,i,j}$	The time required for a Type u UAV to travel from node i to j.
R_1, R_2	Ranges of Type A and B UAVs, respectively.

Table 4.2: Elements of the UAVMPP

Line	Description
4.18	Objective Function, to maximize reward received from performing tasks.
	Structural Constraints
4.19,4.20	Only 1 strike/ISR may be performed at each target.
4.21	Type A UAVs may perform a maximum of 2 strikes.
4.22, 4.23	All UAVs leave and return to the air base.
4.24, 4.25	UAVs may not return to the air base start node, nor leave the return node.
4.26	Conservation of flow: if a UAV visits a target, it must leave.
4.27-4.30	If a Type B UAV visits a node, it must perform ISR at it. Similarly, if a Type A UAV visits a node, it must perform either a strike, an ISR, or both.
4.31-4.33	For Type A UAVs, if a UAV visits the same target sequentially, it must perform both ISR and strike tasks.
	Constraints Subject to Uncertainty
4.34-4.39	Separation constraints, which require enough spacing between sequential arrivals to account for service of the former target and travel to the latter target.
4.40-4.43	Opportunity window constraints, which ensure that all arrival times occur within target opportunity window limitations.
4.44, 4.45	Range constraints, which ensure that UAVs observe their range limitations.
	Decision Variable Specifications
4.46-4.50	Nonnegativity and binary constraints.

Table 4.3: A line by line description of the UAVMPP

$$\max \sum_{i \in N} \Big(\sum_{k \in K} C_{ISR,i} ISR_{k,i} + \sum_{k \in UAV_A} C_{STR,i} STR_{k,i} \Big) \tag{4.18}$$

$$\text{s.t} \quad \sum_{k \in UAV_A} STR_{k,i} \leq 1 \quad \forall i \in T \tag{4.19}$$

$$\sum_{k \in K} ISR_{k,i} \leq 1 \quad \forall i \in T \tag{4.20}$$

$$\sum_{i \in T} STR_{k,i} \leq 2 \quad \forall k \in UAV_A \tag{4.21}$$

$$\sum_{j \in N \setminus 0} x_{k,0,j} = 1 \quad \forall k \in K \tag{4.22}$$

$$\sum_{i \in N \setminus n} x_{k,i,n} = 1 \quad \forall k \in K \tag{4.23}$$

$$x_{k,i,0} = 0 \quad \forall k \in K,\ i \in N \tag{4.24}$$

$$x_{k,n,i} = 0 \quad \forall k \in K,\ i \in N \tag{4.25}$$

$$\sum_{i \in N} x_{k,i,j} = \sum_{i \in N} x_{k,j,i} \quad \forall k \in K,\ j \in T \tag{4.26}$$

$$\sum_{i \in N \setminus j} x_{k,j,i} = ISR_{k,j} \quad \forall k \in UAV_B,\ j \in T \tag{4.27}$$

$$\sum_{i \in N \setminus j} x_{k,j,i} \leq ISR_{k,j} + STR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.28}$$

$$\sum_{i \in N \setminus j} x_{k,j,i} \geq ISR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.29}$$

$$\sum_{i \in N \setminus j} x_{k,j,i} \geq STR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.30}$$

$$x_{k,j,j} \leq STR_{k,j} \quad \forall k \in UAV_A, j \in T \tag{4.31}$$

$$x_{k,j,j} \leq ISR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.32}$$

$$x_{k,j,j} \geq STR_{k,j} + ISR_{k,j} - 1 \quad \forall k \in UAV_A,\ j \in T \tag{4.33}$$

$$w_{ISR,k,i} + S_i^2 x_{k,i,j} + D_{2,i,j} x_{k,i,j} \leq w_{ISR,k,j} + (1 - x_{k,i,j})\, M \quad \forall k \in UAV_B,\ \forall i,j \in N | i \neq j \tag{4.34}$$

$$w_{ISR,k,i} + S_{ISR,i}^1 ISR_{k,i} + D_{1,i,j} x_{k,i,j} \leq w_{STR,k,j} + (2 - STR_{k,j} - x_{k,i,j})M \quad \forall k \in UAV_A,\ i,j \in N | i \neq j \tag{4.35}$$

$$w_{ISR,k,i} + S_{ISR,i}^1 ISR_{k,i} + D_{1,i,j} x_{k,i,j} \leq w_{ISR,k,j} + (2 - ISR_{k,j} - x_{k,i,j})M \quad \forall k \in UAV_A,\ i,j \in N | i \neq j \tag{4.36}$$

$$w_{STR,k,i} + S_{STR,i}^1 STR_{k,i} + D_{1,i,j} x_{k,i,j} \leq w_{STR,k,j} + (2 - STR_{k,j} - x_{k,i,j})M \quad \forall k \in UAV_A,\ i,j \in N | i \neq j \tag{4.37}$$

$$w_{STR,k,i} + S_{STR,i}^1 STR_{k,i} + D_{1,i,j} x_{k,i,j} \leq w_{ISR,k,j} + (2 - ISR_{k,j} - x_{k,i,j})M \quad \forall k \in UAV_A,\ i,j \in N | i \neq j \tag{4.38}$$

$$w_{ISR,k,i} + S_{ISR,i}^1 x_{k,i,i} \leq w_{STR,k,i} + (1 - x_{k,i,i})M \quad \forall k \in UAV_A,\ i, \in N \tag{4.39}$$

$$A_i ISR_{k,i} \leq w_{ISR,k,i} \quad \forall k \in K,\ i \in T \tag{4.40}$$

$$A_i STR_{k,i} \leq w_{STR,k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.41}$$

$$w_{ISR,k,i} \leq B_i ISR_{k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.42}$$

$$w_{STR,k,i} \leq B_i STR_{k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.43}$$

$$\sum_{i \in N}(S_i^2 ISR_{k,i} + \sum_{j \in N} D_{2,i,j} x_{k,i,j}) \leq R_2 \quad \forall k \in UAV_B \tag{4.44}$$

$$\sum_{i \in N}(S_{ISR,i}^1 ISR_{k,i} + S_{STR,i}^1 STR_{k,i} + \sum_{j \in N} D_{1,i,j} x_{k,i,j}) \leq R_1 \ \forall k \in UAV_A \tag{4.45}$$

$$x_{k,i,j} \in \{0,1\} \quad \forall k \in K, \ \forall i,j \in N \tag{4.46}$$

$$ISR_{k,i} \in \{0,1\} \quad \forall k \in K, \ \forall i \in N \tag{4.47}$$

$$STR_{k,i} \in \{0,1\} \quad \forall k \in UAV_A, \ \forall i \in N \tag{4.48}$$

$$w_{STR,k,i} \geq 0 \quad \forall i \in N, \ \forall k \in UAV_A \tag{4.49}$$

$$w_{ISR,k,i} \geq 0 \quad \forall i \in N, \ \forall k \in K \tag{4.50}$$

As mentioned, we are especially interested in obtaining a formulation for the UAVMPP that incorporates uncertainty. Many studies have already investigated the use of heuristics in solving VRP-type problems. While these heuristics are attractive in that they generally find solutions quickly, the solutions are not guaranteed to be optimal and generally do not incorporate uncertainty. While Dynamic Programming is capable of accounting for the uncertainty in the problem, we determine that it is not computationally feasible for the UAVMPP. Though Branch-and-Bound is also computationally intensive, market-available solvers utilize algorithms that are refined and capable of operating much more efficiently. In addition, the scope of the UAVMPP laid out in Section 3 dictates that the problems we solve will contain relatively small numbers of targets and UAVs. The availability of efficient Branch-and-Bound algorithms, along with our problem size leads us to believe that the Branch-and-Bound algorithm will suffice. Furthermore, with the emergence of Robust Optimization techniques, it is now possible to incorporate uncertainty into plans that are solved using Branch-and-Bound algorithms. Among the techniques available, we feel that the Bertsimas-Sim formulation is the most promising, in that it does not add complexity to the underlying problem and allows the user control over the amount of protection to give to the solution. Moreover, the formulation is relatively new, and has yet to receive extensive testing on real applications. Our use of the formulation for the UAVMPP provides a good testbed with which to measure its performance on a realistic problem.

The formulation we present makes a few modifications to the Bertsimas-Sim for-

mulation. For ease of presentation, the z and p dual variables are replaced by the Greek θ and ρ, respectively. This change makes it easier to distinguish the dual variables from the original problem variables, which assume letters from the English alphabet. The $\mathbf{y}$ variable is not necessary, as our decision variables are limited to nonnegative values, and it is removed. Similarly, ρ is removed in constraints for which $|J| = 1$, as they always assume a value of zero in these instances. Finally, we replace Γ with the parameter *prot*, which is defined as the fraction of the uncertain coefficients in a constraint that the formulation protects against. In each constraint, *prot* is multiplied by the number of uncertain coefficients in the constraint, with this number being represented by J_1 in the objective function of our formulation (as the number of uncertain coefficients in the objective function is dependent on the size of the problem. $J_1 = |N|(|K| + |UAV_A|)$). This transformation simplifies coding, and allows us to specify a global protection parameter for runs in which we do not vary protection by constraint.

In the course of our research, we experimented with several different formulation versions, but only performed testing on one of them. The versions that are not included in the testing are included as appendices. We present the test-model of the Robust UAVMPP formulation below. The robust formulation builds off of the deterministic one presented earlier. It has the two new decision variables, ρ and θ, mentioned earlier. Because a θ must exist for every uncertain constraint, and a ρ for every uncertain coefficient in each constraint, we use superscripts and subscripts to differentiate among these dual variables. In our formulation, the superscripts represent the constraint set (each line of the the formulation actually represents a set of constraints of identical structure) that the variable belongs to, and the subscripts indicate the particular constraint within the set. Because some constraints contain multiple uncertain coefficients, the subscripts for ρ also indicate which of these coefficients the ρ belongs to.

Hat values exist for all of the uncertain coefficients (C,S,D,A,B,R), and are indicated with a hat mark above the letter for the coefficient. The objective function on line (4.51) is modified from the original problem, as are the constraints on lines

(4.68) through (4.79), which are subject to uncertainty. While the formulation in [12] does not have protection applied to its objective function, it does so for reasons of simplicity, and assumes that the user will move the objective function to the "A" matrix if it contains uncertainty. In addition, lines (4.80) through (4.99) are added as dual constraints of the Bertsimas-Sim formulation. Table 4.4 associates the dual constraints with the primal constraints they correspond to.

Dual Constraints	
4.80-4.81	Objective function dual constraints
4.82-4.88	Separation constraint duals.
4.89-4.92	Opportunity window constraint duals.
4.93-4.99	Range constraint duals.

Table 4.4: Dual constraints of the Robust UAVMPP formulation

$$\max \sum_{i\in N}(\sum_{k\in K} C_{ISR,i} ISR_{k,i} + \sum_{k\in UAV_A} C_{STR,i} STR_{k,i}) - \tag{4.51}$$

$$J_1(prot)\ \theta_1^1 - \sum_{i\in N}(\sum_{k\in K} \rho_{k,i}^1 + \sum_{k\in UAV_A} \rho s_{k,i}^1) \tag{4.52}$$

$$\text{s.t} \sum_{k\in UAV_A} STR_{k,i} \leq 1 \quad \forall i \in T \tag{4.53}$$

$$\sum_{k\in K} ISR_{k,i} \leq 1 \quad \forall i \in T \tag{4.54}$$

$$\sum_{i\in T} STR_{k,i} \leq 2 \quad \forall k \in UAV_A \tag{4.55}$$

$$\sum_{j\in N\setminus 0} x_{k,0,j} = 1 \quad \forall k \in K \tag{4.56}$$

$$\sum_{i\in N\setminus n} x_{k,i,n} = 1 \quad \forall k \in K \tag{4.57}$$

$$x_{k,i,0} = 0 \quad \forall k \in K,\ i \in N \tag{4.58}$$

$$x_{k,n,i} = 0 \quad \forall k \in K,\ i \in N \tag{4.59}$$

$$\sum_{i\in N} x_{k,i,j} = \sum_{i\in N} x_{k,j,i} \quad \forall k \in K,\ j \in T \tag{4.60}$$

$$\sum_{i\in N\setminus j} x_{k,j,i} = ISR_{k,j} \quad \forall k \in UAV_B,\ j \in T \tag{4.61}$$

$$\sum_{i\in N\setminus j} x_{k,j,i} \leq ISR_{k,j} + STR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.62}$$

$$\sum_{i\in N\setminus j} x_{k,j,i} \geq ISR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.63}$$

$$\sum_{i\in N\setminus j} x_{k,j,i} \geq STR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.64}$$

$$x_{k,j,j} \leq STR_{k,j} \quad \forall k \in UAV_A, j \in T \tag{4.65}$$

$$x_{k,j,j} \leq ISR_{k,j} \quad \forall k \in UAV_A,\ j \in T \tag{4.66}$$

$$x_{k,j,j} \geq STR_{k,j} + ISR_{k,j} - 1 \quad \forall k \in UAV_A,\ j \in T \tag{4.67}$$

$$w_{ISR,k,i} + S_i^2 x_{k,i,j} + D_{2,i,j} x_{k,i,j} + 2(prot)\theta_{k,i,j}^2 + \sum_{u \in \{1,2\}} \rho_{u,k,i,j}^2 \leq w_{ISR,k,j} + (1 - x_{k,i,j})\, M \tag{4.68}$$

$$\forall k \in UAV_B,\ \forall i,j \in N | i \neq j$$

$$w_{ISR,k,i} + S_{ISR,i}^1 ISR_{k,i} + D_{1,i,j} x_{k,i,j} + 2(prot)\theta_{k,i,j}^2 + \sum_{u \in \{1,2\}} \rho_{u,k,i,j}^2 \leq w_{STR,k,j} + (2 - STR_{k,j} - x_{k,i,j})M \tag{4.69}$$

$$\forall k \in UAV_A,\ i,j \in N | i \neq j$$

$$w_{ISR,k,i} + S_{ISR,i}^1 ISR_{k,i} + D_{1,i,j} x_{k,i,j} + 2(prot)\theta_{k,i,j}^2 + \sum_{u \in \{1,2\}} \rho_{u,k,i,j}^2 \leq w_{ISR,k,j} + (2 - ISR_{k,j} - x_{k,i,j})M \tag{4.70}$$

$$\forall k \in UAV_A,\ i,j \in N | i \neq j$$

$$w_{STR,k,i} + S_{STR,i}^1 STR_{k,i} + D_{1,i,j} x_{k,i,j} + 2(prot)\theta_{k,i,j}^3 + \rho_{1,k,i,j}^3 + \rho_{2,k,i,j}^3 \leq w_{STR,k,j} + (2 - STR_{k,j} - x_{k,i,j})M \tag{4.71}$$

$$\forall k \in UAV_A,\ i,j \in N | i \neq j$$

$$w_{STR,k,i} + S_{STR,i}^1 STR_{k,i} + D_{1,i,j} x_{k,i,j} + 2(prot)\theta_{k,i,j}^3 + \rho_{1,k,i,j}^3 + \rho_{2,k,i,j}^3 \leq w_{ISR,k,j} + (2 - ISR_{k,j} - x_{k,i,j})M \tag{4.72}$$

$$\forall k \in UAV_A,\ i,j \in N | i \neq j$$

$$w_{ISR,k,i} + S_{ISR,i}^1 x_{k,i,i} + (prot)\theta_{k,i}^4 \leq w_{STR,k,i} + (1 - x_{k,i,i})M \tag{4.73}$$

$$\forall k \in UAV_A,\ i, \in N$$

$$A_i ISR_{k,i} + (prot)\theta_{k,i}^5 \leq w_{ISR,k,i} \quad \forall k \in K,\ i \in T \tag{4.74}$$

$$A_i STR_{k,i} + (prot)\theta_{k,i}^6 \leq w_{STR,k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.75}$$

$$w_{ISR,k,i} + (prot)\theta_{k,i}^7 \leq B_i ISR_{k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.76}$$

$$w_{STR,k,i} + (prot)\theta_{k,i}^8 \leq B_i STR_{k,i} \quad \forall k \in UAV_A,\ i \in T \tag{4.77}$$

$$\sum_{i \in N}(S_i^2 ISR_{k,i} + \sum_{j \in N} D_{2,i,j} x_{k,i,j}) + 3(prot)\theta_k^9 + \sum_{i \in N} \rho_{k,i}^9 + \sum_{i \in N, j \in N} \rho a_{k,i,j}^9 + \rho b_k^9 \leq R_2 \quad \forall k \in UAV_B \tag{4.78}$$

$$\sum_{i \in N}(S_{ISR,i}^1 ISR_{k,i} + S_{STR,i}^1 STR_{k,i} + \sum_{j \in N} D_{1,i,j} x_{k,i,j}) \leq -4(prot)\theta_k^{10} - \sum_{i \in N}(\rho_{k,i}^{10} + \rho a_{k,i}^{10}) - \sum_{i \in N, j \in N} \rho b_{k,i,j}^{10} - \rho c_k^{10} + R_1 \quad \forall k \in UAV_A \tag{4.79}$$

$$\hat{C}_{ISR,i} ISR_{k,i} \leq \theta^1 + \rho_{k,i}^1 \quad \forall i \in N,\ k \in K \tag{4.80}$$

$$\hat{C}_{STR,i} STR_{k,i} \leq \theta^1 + \rho s_{k,i}^1 \quad \forall i \in N,\ k \in UAV_A \tag{4.81}$$

$$\hat{S^2}_i ISR_{k,i} \leq \theta_{k,i,j}^2 + \rho_{1,k,i,j}^2 \quad \forall i,j \in N,\ k \in UAV_B \tag{4.82}$$

$$\hat{D}_{2,i,j} x_{k,i,j} \leq \theta_{k,i,j}^2 + \rho_{2,k,i,j}^2 \quad \forall i,j \in N\ ,\ k \in UAV_B \tag{4.83}$$

$$\hat{S^1}_{ISR,i} ISR_{k,i} \leq \theta_{k,i,j}^2 + \rho_{1,k,i,j}^2 \quad \forall i,j \in N,\ k \in UAV_A \tag{4.84}$$

$$\hat{D}_{1,i,j} x_{k,i,j} \leq \theta_{k,i,j}^2 + \rho_{2,k,i,j}^2 \quad \forall i,j \in N\ ,\ k \in UAV_A \tag{4.85}$$

$$\hat{S^1}_{STR,i} STR_{k,i} \leq \theta_{k,i,j}^3 + \rho_{1,k,i,j}^3 \quad \forall i,j \in N,\ k \in UAV_A \tag{4.86}$$

$$\hat{D}_{1,i,j}x_{k,i,j} \leq \theta^3_{k,i,j} + \rho^3_{2,k,i,j} \quad \forall i,j \in N\ ,\ k \in UAV_A \tag{4.87}$$

$$\hat{S^1}_{ISR,i}ISR_{k,i} \leq \theta^4_{k,i} + \rho^4_{k,i} \quad \forall i \in N,\ k \in UAV_A \tag{4.88}$$

$$\hat{A}_i ISR_{k,i} \leq \theta^5_{k,i} + \rho^5_{k,i} \quad \forall i \in N,\ k \in K \tag{4.89}$$

$$\hat{A}_i STR_{k,i} \leq \theta^6_{k,i} + \rho^6_{k,i} \quad \forall i \in N,\ k \in UAV_A \tag{4.90}$$

$$\hat{B}_i ISR_{k,i} \leq \theta^7_{k,i} + \rho^7_{k,i} \quad \forall i \in N,\ k \in K \tag{4.91}$$

$$\hat{B}_i STR_{k,i} \leq \theta^8_{k,i} + \rho^8_{k,i} \quad \forall i \in N\ \forall k \in UAV_A \tag{4.92}$$

$$\hat{S^2}_i ISR_{k,i} \leq \theta^9_k + \rho^9_{k,i} \quad \forall k \in UAV_B,\ i \in N \tag{4.93}$$

$$\hat{D}_{2,i,j}x_{k,i,j} \leq \theta^9_k + \rho a^9_{k,i,j} \quad \forall k \in UAV_B,\ i,j \in N \tag{4.94}$$

$$\hat{R}_2 \leq \theta^9_k + \rho b^9_k \quad \forall k \in UAV_B \tag{4.95}$$

$$\hat{S^1}_{ISR,i}ISR_{k,i} \leq \theta^{10}_k + \rho^{10}_{k,i} \quad \forall k \in UAV_A,\ i \in N \tag{4.96}$$

$$\hat{S^1}_{STR,i}STR_{k,i} \leq \theta^{10}_k + \rho a^{10}_{k,i} \quad \forall k \in UAV_A,\ i \in N \tag{4.97}$$

$$\hat{D}_{1,i,j}x_{k,i,j} \leq \theta^{10}_k + \rho b^{10}_{k,i,j} \quad \forall k \in UAV_A,\ i,j \in N \tag{4.98}$$

$$\hat{R}_1 \leq \theta^{10}_k + \rho c^{10}_k \quad \forall k \in UAV_A \tag{4.99}$$

$$x_{k,i,j} \in \{0,1\} \quad \forall k \in K,\ \forall i,j \in N \tag{4.100}$$

$$ISR_{k,i} \in \{0,1\} \quad \forall k \in K,\ \forall i \in N \tag{4.101}$$

$$STR_{k,i} \in \{0,1\} \quad \forall k \in UAV_A,\ \forall i \in N \tag{4.102}$$

$$w_{STR,k,i} \geq 0 \quad \forall i \in N,\ \forall k \in UAV_A \tag{4.103}$$

$$w_{ISR,k,i} \geq 0 \quad \forall i \in N,\ \forall k \in K \tag{4.104}$$

$$\rho, \theta \geq 0 \tag{4.105}$$

Chapter 5

Tests and Analysis

This chapter covers the testing portion of the research. In our testing, we seek to establish the quality of our formulation, and to discern the effects of increased protection on the performance of solutions. Unfortunately, because our system is hypothetical, it is impossible for us to test the performance of this formulation in "real life." Instead, we develop a simulation test bed that approximates a real life environment. Using this test bed, we perform experiments that address specific hypotheses regarding the performance of the formulation and solutions. In order to test performance, we develop metrics to use in comparing the simulation results of different solutions. The topics of the chapter are presented in the following order:

1. Metrics
2. Test Bed
3. Hypotheses
4. Experiments and write-ups
5. Summary of findings

5.1 Metrics

In our testing of the formulation, we desire to compare solutions that are protected by the formulation with the "nominal" (unprotected) solutions of the UAVMPP. To this

end, we require a set of metrics with which to quantify the solution quality. Values for these metrics can be obtained through simulation, and these metrics can then be used as a basis of comparison.

In most optimization problems, solutions are compared on the basis of their objective function yield. For our purpose, however, a simple comparison of objective function value is inadequate. Because added protection, in reducing the feasible region, involves a tradeoff with optimality, the nominal case will no doubt always have the best apparent objective function value ($Obj(nominal) \geq Obj(protected)$ for any level of protection). This value, however, might never be realized because of constraint violations that occur when the realized coefficient values of the simulation differ from those planned for by the nominal solution. In this case, substantial differences can exist between this realized simulation objective value, or "simulation value", and the expected objective function value. Moreover, it is conceivable that constraint violations bear costs that are not reflected in the objective function, but which should still be considered in comparing solutions.

It is apparent, then, that constraint violation might be another metric of interest. Constraint violation, itself, can be subdivided into more metrics, as it is possible to record violations by the constraint type. This might be desirable when differences in the importance of the constraints are perceived, making violations of some constraints more serious than violations of others. Violations in certain constraints, for instance, might bear more implications in terms of plan executability and value gained than in others. We now relate this to our problem.

In our formulation, four types of expressions contain uncertain coefficients: the objective function, the Separation Constraints, the Opportunity Window Constraints, and the Range Constraints. The objective is a maximization function that does not contain an inequality, and is therefore not subject to violation. The Separation Constraints (equations 4.68 to 4.73 of Section 4.4) dictate that routes must provide enough spacing between the scheduled arrival times at contiguous targets to allow for service of the first target and travel to the subsequent target, and are of the form $w_i + S_i + D_{ij} \leq w_j$. The Opportunity Window Constraints (equations 4.74 to 4.77)

Uncertain Constraints

Constraint Name	Form	Violation Reference Name
Separation	$w_i + S_i + D_{ij} \leq w_j$	Type 1
Opportunity Window	$A_i \leq w_i \leq B_i$	Type 2
Range	$\sum_{i \in N}(S_i y_i) + \sum_{i,j \in N}(D_{ij} x_{ij}) \leq R$	Type 3

Table 5.1: The three types of constraints subject to uncertainty

dictate that arrival times occur within the opportunity windows specified at each target, and are of the form $A_i \leq w_i \leq B_i$. The Range Constraints (equations 4.78 to 4.79) ensure that the range limitations of the vehicles are observed, and are of the form $\sum_{i \in N}(S_i y_i) + \sum_{i,j \in N}(D_{ij} x_{ij}) \leq R$ (see Table 5.1).

In terms of their impact on the value received from a plan, Opportunity Window Constraint violations, which we refer to as Type 2 violations, are serious. We assume that a target is not available outside of its opportunity window, and that no value is gained by UAVs that arrive at the target outside of this window. Violations of the Separation Constraints, the Type 1 violations, do not bear the same implications. A violation among these constraints does not necessarily indicate that the target in question cannot be visited, but only that this target cannot be visited at the planned time (w). As long as a time within the target opportunity window remains feasible for the plan, this target can still be visited. That said, breaking a scheduled target arrival time can certainly carry a cost with it. The costs might be internal to our system: they might preclude missions that require a coordinated effort among several vehicles, or might cause the same UAV to miss subsequent tasks assigned to it. The costs might also be external to the system: the departure might create conflicts with external plans such as the refueling schedule, or might stall a ground operation that is contingent on completion of the UAV missions. Quantifying the cost of a Type 1 violation is a nontrivial task. One should note that while arriving to a target earlier than planned does constitute a departure from the plan, it does not entail a constraint

violation. Intuitively, a UAV that arrives earlier than scheduled can loiter in the area until the scheduled arrival time, and then proceed with the mission. Also, missing a target arrival time does not imply that the remaining targets in the UAV's route are unattainable; the UAV can proceed with the schedule and attempt to visit the remaining targets.

Type 3 (Range Constraint) violations are perhaps even more serious than Type 2 violations, as they involve the loss of a UAV (a UAV that is extended beyond its range will crash). It is arguable that rewards should still be tabulated for all of the targets successfully prosecuted by the UAV prior to the point at which UAV range is violated. It is also foreseeable, however, that the penalty incurred from the loss of the UAV would exceed all of the reward obtained from the missions it accomplished. It is expected, however, that an operator would anticipate a range violation and actively try to avoid it. We suggest an algorithm for doing so in the test bed section below.

Because of the varying implications of violations among the different constraint types within our problem, it is not sufficient to track constraint violation as a single metric. Rather, violations should be recorded by their type. However, just as relying solely on the objective function value as a metric is inadequate, so also it is inadequate to consider only these constraint violations when comparing solutions. If the quality of a solution were based solely on constraint violation, the solution of doing nothing would be optimal, as it could never violate the constraints of our formulation. This would hardly be optimal, assuming that the primary purpose of the plan is to maximize the value that is measured by the objective function. It seems, then, that a tradeoff exists between objective function value and constraint violation: improving the one generally comes at the cost of deteriorating the other. Even so, it is possible for plans with identical objective function values to experience differing constraint violations, as it is possible for plans with a similar vulnerability to constraint violation to have differing objective function values. These plans appear identical if only one of the metrics is considered, but, we argue, are not identical, and this further establishes the need for both objective function value and constraint violation to be considered as metrics.

In addition to objective function value and constraint violation, we suggest the earliest time in the plan at which violations occur might be an additional metric of interest. If violations do occur, it is reasonable to assume that having them occur later in the plan is preferable to earlier, as more of the plan is executable as scheduled. Furthermore, the amount by which a constraint is violated might be of interest, as the gravity of the violation might depend upon this value. In terms of our problem, this corresponds to the amount of time by which the UAV missed a planned arrival time or opportunity window, or the amount of time that a UAV exceeded its range.

When all of the realizations are tabulated, and the values for the metrics stored, we calculate both average values and variances to the metrics. While knowledge of the means for the metrics is important, it does not provide a complete view of the results of the test bed runs. Mean simulation value is still subject to the argument of attainability, which caused us to move from considering objective function value to simulation value. If, however, variances about these mean values are considered as well, we can develop a better sense for the likelihood of achieving the particular metric. It is assumed that a lower variance is preferable to a higher one, and that a solution with lower mean simulation value could still be preferable to one with higher mean value if it has a lower variance about this mean. We list the metrics that have thus far been mentioned.

Metrics

- Objective function value
- Average realized simulation value
- Average Type 1 violation
- Average Type 2 violation
- Average Type 3 violation
- Average departure times for all violation types
- Variances for all values

Another metric we track is solve time. Our model must be computationally tractable in order for it to have any practical value. While the model is posed as a planner, and as such would not be faced with the time limitations of a real-time system, the model should nonetheless arrive at a near optimal solution in a reasonable time, given the computational power available to users. Furthermore, a real-time closed-loop feedback system is a natural extension to this research and would be highly desirable of a system that would eventually be implemented. Thus, solve time is certainly of interest to us, and we will keep track of it as a metric.

We conclude this section with the acknowledgement that a unified metric that could encapsulate all of these listed metrics would be desirable. A single metric would make comparison of the different plans a simple task, and plans that yielded higher values for this metric would be clearly and undeniably better than those with lower values. This being said, we nonetheless decide not to use such a metric. Unifying the current metrics would involve assigning penalties to violations, the amount by which the constraints were violated, and the times at which these violations occurred. We feel that in doing so we would be making assumptions that could compromise the general applicability of our results.

5.2 Test Bed

This section describes the software and hardware of the test bed, as well as the scenarios used in the testing. The test bed is composed of a series of Matlab scripts that call the Dash Optimization XPress-MP Solver v2005 via console commands. XPress finds the solution to the UAVMPP by first using the Simplex Method to solve for the linear relaxation of the problem, and then finding the optimal integer solution via its branch-and-bound algorithm. Once the solution is obtained, it is tested through simulation. The Matlab random number generator is used to generate data instances for all of the uncertain coefficients. The test bed allows for the use of either a normal or uniform distribution to find instances within the coefficient ranges specified by the coefficient mean values and hat values. Two Matlab scripts

evaluate the XPress-developed solutions with the randomly-generated data instances, and provide values for the metrics that were presented in Section 5.1. The two simulations use slightly different logic in stepping through the plans, as is presented later in this section.

This process is repeated for each plan that is tested. Each scenario is tested at eleven levels of protection: from 0 to 1 in 10% increments, with these values representing the percent of the total number of uncertain coefficients in each constraint that receive protection. Each solution is evaluated with 1000 instances of the uncertain coefficients. After testing is complete and results are recorded, output graphs are generated by a separate Matlab script. The entire process is depicted in Figure 5-1 below, which presents the test bed functions along with corresponding descriptions, and shows the flow and structure between them.

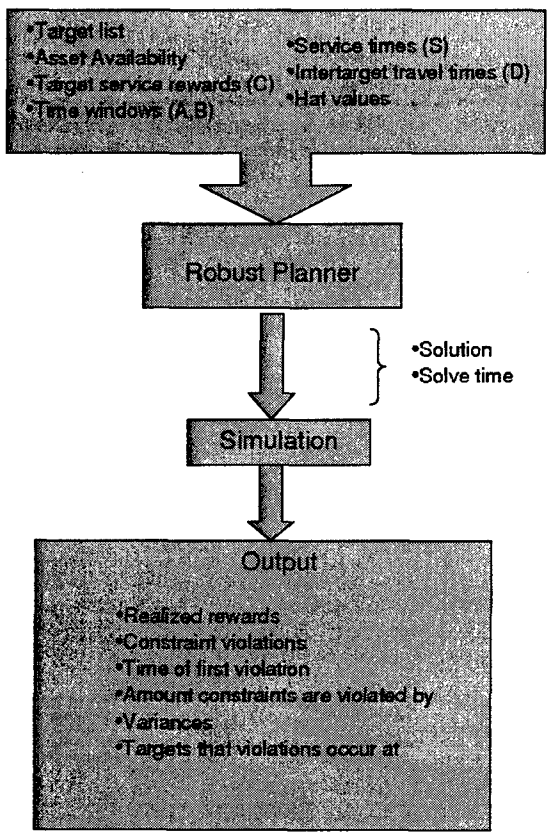


Figure 5-1: Overview of simulation with parameter flow

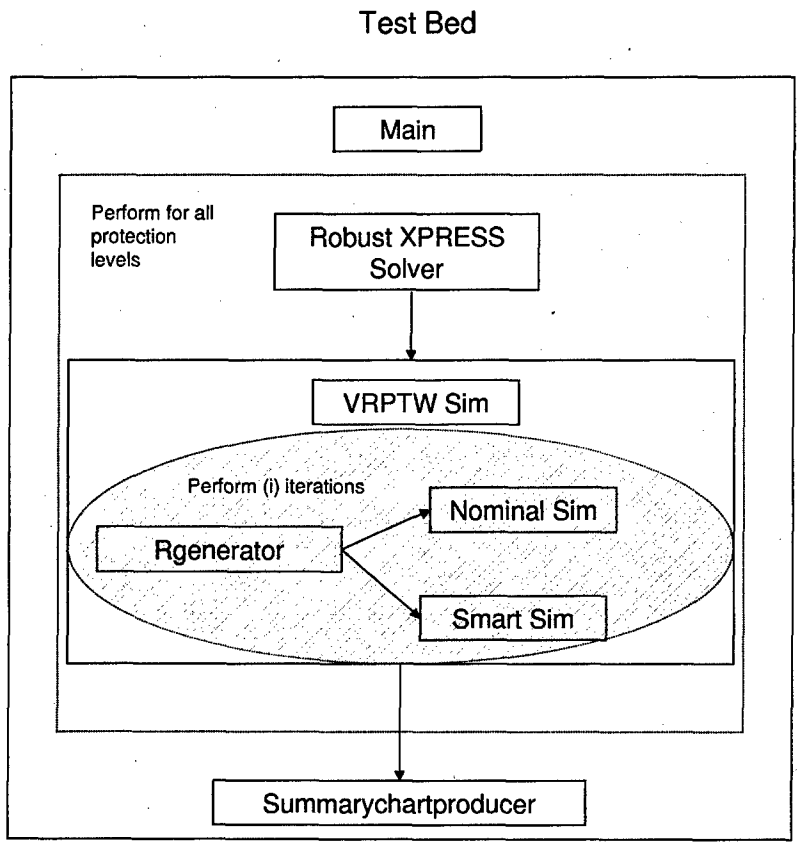


Figure 5-2: Simulation test bed

Two simulations, each inspired by the discussion in Section 5.1, are performed on the test bed (see Figure 5-2). The simulations progress through the plans in a way that approximates how we believe an intelligent controller or Combat Operations Division (COD) would. Both simulations step through the tasks of each UAV chronologically, receiving benefit based on whether or not the tasks can be accomplished under the realized data. Counts of the three violation types are maintained for each simulation run. In addition, the earliest time of each violation type is maintained for each run, as is the amount by which Separation Constraints are violated.

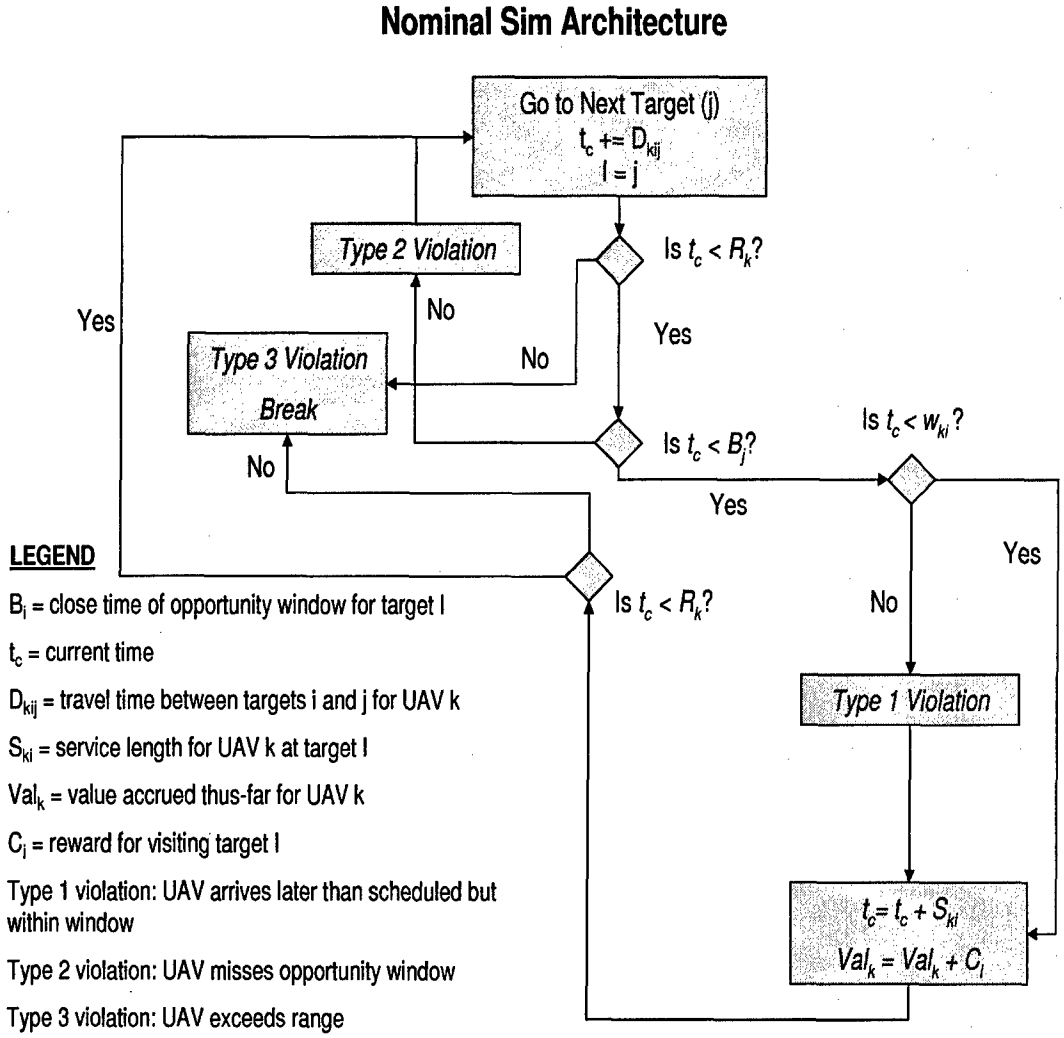


Figure 5-3: Simulation architecture for Nominal Sim

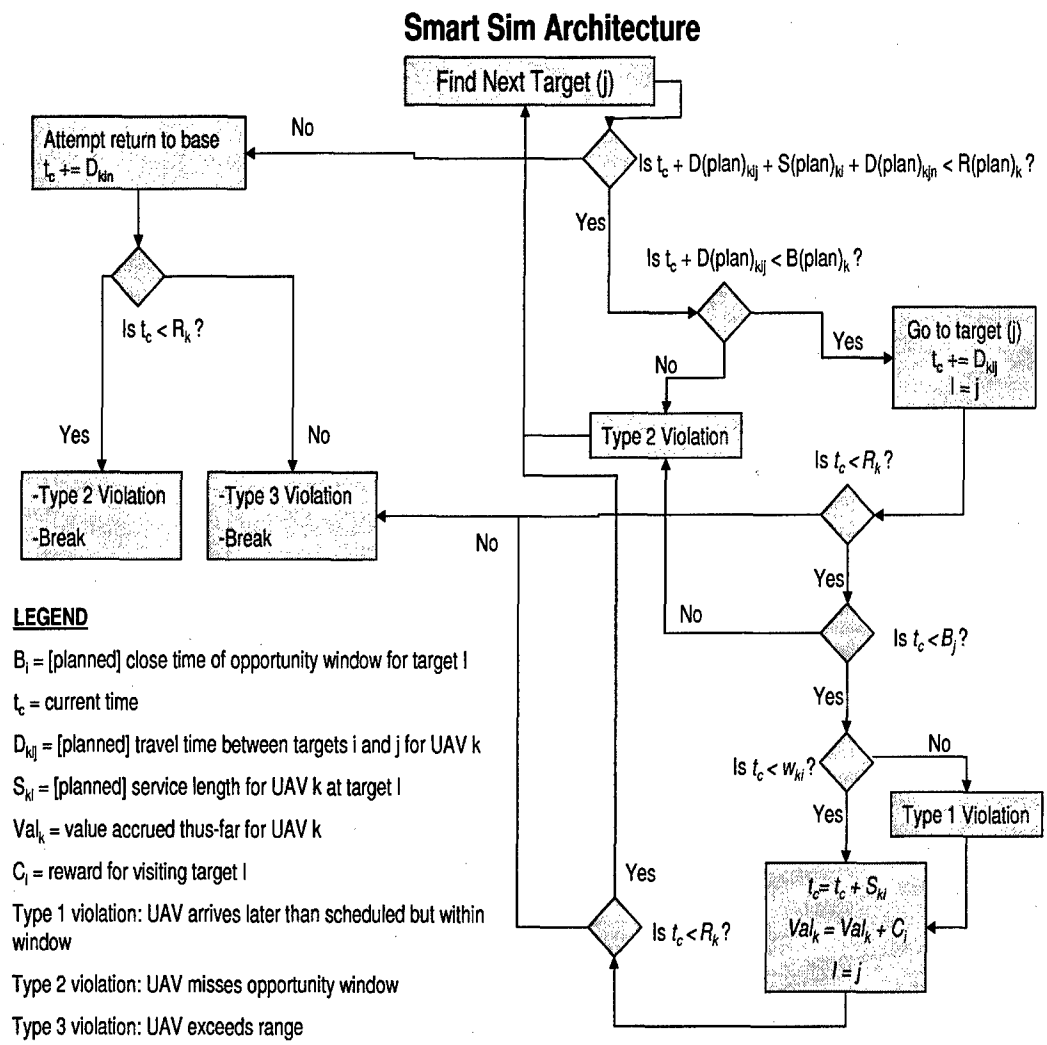


Figure 5-4: Simulation architecture for Smart Sim

5.2.1 Simulation Architecture

Before describing the logic of the two simulations, we first present our rationale in choosing to step through the plans in a chronological manner, instead of simultaneously testing for violations in all the constraints. Constraints, and consequently constraint violations, are dependent on one another. Even the formulation decision variables, which often appear together in the constraints, are dependent (the Separation Constraints introduce the dependency between the target arrival times of a UAV). If we were to test for violations in all constraints simultaneously, these dependencies would not be properly accounted for. Because of the dependencies, a violation in one constraint can influence violations among other constraints. These violations can build upon one another in the UAVMPP, and tend to have a snowball effect, where small departures from the expected value compound to create significant problems later on in the planned routes. For example, let us assume that it takes a particular UAV longer than expected to travel to its first target and service it. This delays the arrival time at that target, and could result in a delay at the next target as well, prompting a Type 1 and potentially a Type 2 or 3 violation. While the initial departure from the plan was small, it had the potential to compound and cause other, potentially more significant violations. In order to properly account for the influence that violations have on tasks occurring later in the UAV routes, it is necessary to step through the UAV routes in a chronological manner, and view the realizations as they would be viewed by the UAVs.

We now explain the differences in logic between the two simulations. The first, which we refer to as a "Nominal" Simulation, mimics a UAV controller that does not attempt to foretell whether a UAV will arrive at a target in a timely manner before allowing it to depart for the target. If the UAV arrives late to the target, a Type 2 violation is raised and no value is gained from the task assigned at that target. The UAV in question attempts to perform the next task on its itinerary, and has "wasted" the travel time to the present target (Figure 5-3).

The second simulation is referred to as the "Smart" Sim, as it allows the UAV con-

troller to make slight perturbations from the plan based on the data realizations that it has already encountered. The simulation algorithm checks, for instance, whether a UAV is expected to arrive within the close of the opportunity window specified at the next target before allowing it to leave for the target. If, based on planned data, the UAV does not appear capable of arriving at the target within the specified time limit, the simulation instead conserves the UAV's range and directs it to visit the next target in the plan. The Smart Sim also checks whether a UAV will be capable of visiting the next target, performing the scheduled task, and returning to the airbase within the UAV range limitations. If this does not appear to be possible, the simulation instead orders the UAV to return to the airbase. While the controller makes its perturbation decisions based on nominal data, the plan nonetheless encounters data realizations. As a result, the controller may both make the decision for a UAV to skip a target when in actuality it could have visited it, and may determine for a UAV to visit a target, only to find that the UAV still did not satisfy the opportunity window or range limitations. Figure 5-4 illustrates the logic used by the Smart Sim.

Hardware used in the testing was a Pentium IV system with dual 1.3Ghz processors and 1 gigabyte of 2 Dimm SDRAM. In the following section, we discuss the scenarios that are tested throughout the experimentation.

5.2.2 Scenarios

Throughout the experimentation, we test the formulation and the plans produced by it under different problem circumstances, or scenarios. These scenarios contain all parameters needed to define a separate instance of our UAVMPP, and exist as data files that are input into the XPress solver and Matlab scripts. The required parameters correspond to the sets and constants defined in the formulation section (Section 4). In an effort to make the results of our testing more tangible, we based all scenarios on a hypothetical conflict with a fictitious nation. The conflict occurs over the airspace of the nation, and simulates an invasion of friendly forces into the nation. The scenarios designate the locations of the airbase and targets assigned to the UAV unit for a particular ATO cycle. As indicated in Chapter 3, locations are

	Target Types		
	SAM Site	**Mobile Artillery Unit**	**Leadership**
Reward (Strike/ISR)	100/60	50/30	200/120
Service-Time ISRA/ISRB/Strike	0.5/0.65/0.8	2/2.6/3.2	3/3.9/4.8
Opportunity-Window (hrs)	60	20	10
Hat-Multiplier,m ($\hat{a} = m\bar{a}$)	0.1	0.15	0.3

Table 5.2: Overview of target types

specified as x and y coordinates, but are converted into the inter-target travel-time approximations that the formulation uses via straight line distances and assumptions about UAV speeds.

In all scenarios, three types of targets exist: Surface to Air Missile (SAM) Sites, Mobile Artillery Units (MAU), and Leadership targets. These target types define the coefficient values associated with the targets. They define expected values for the reward for task completion at a target, target service period, and the length of the target opportunity window. They also specify the uncertainty ranges that lie about these expected values. SAM sites, being stationary, have smaller expected service period times, because the portion of the service period allotted to finding the target is small. They yield a fairly high reward, have large opportunity windows, and have relatively small uncertainties surrounding these values. MAUs are less threatening to friendly aircraft but also more elusive. They yield less reward, have shorter opportunity windows, longer service times, and have a higher level of uncertainty. Leadership targets are the most valuable, but also the most difficult to successfully prosecute. They have the highest reward, but the smallest opportunity windows, longest service times, and most uncertainty. Figure 5-5 depicts the target-airbase configuration of a sample scenario, and Figure 5.2.2 summarizes the coefficient values associated with each target type. We provide overviews of the specific scenarios and parameter values as we use them in each of the experiments.

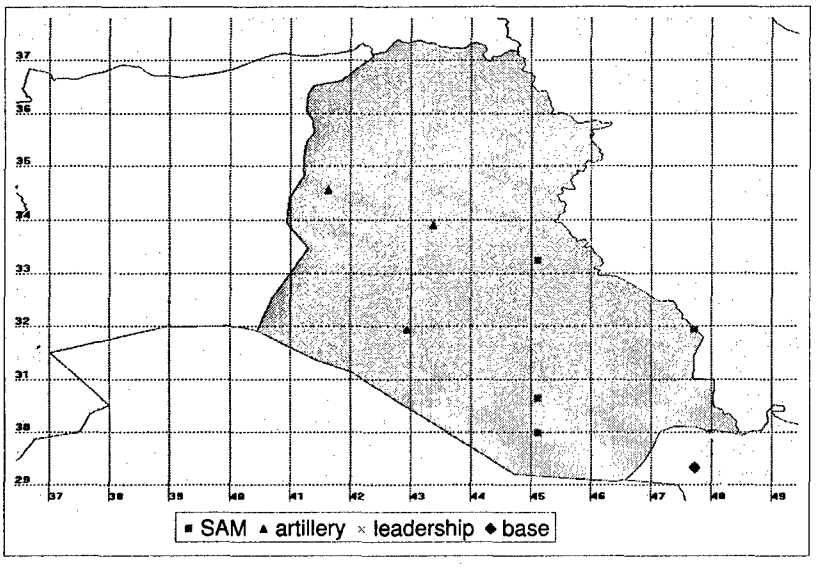


Figure 5-5: Example map of scenarios

5.3 Hypotheses

As mentioned, our interest in testing is to investigate the quality of the protected solutions in relation to nominal ones. Our expectation is that, on average, the protected solutions will perform better across all of the metrics established in Section 5.1. Namely, we believe that realized values for protected solutions will, on average, outperform those of the nominal solutions; there will be, on average, fewer violations among all constraint types; the average earliest time of departure will increase for all departure types; average lateness will decrease; and average variance will decline for all metrics. These expectations motivate the experiments, and form the "base" hypotheses that are tested. We now enumerate the Base Hypotheses used in our experimentation.

Base Hypotheses

1. Increasing protection will cause the average plan to be executable for longer periods of time before experiencing a violation

2. Increasing protection causes plans to experience fewer constraint violations on

average

3. Increasing protection causes violated constraints to be violated by less, on average

4. Increasing protection increases the expected plan value to the user

5. Increasing protection causes the expected return (objective function value) to more closely match the achieved (realized) value

6. Average variance on all metrics decreases as protection is increased

5.3.1 Beliefs on Performance

Our basic belief is that protected solutions will begin to outperform nominal ones as constraint violation increases among the realizations of the nominal solution. While protected solutions utilize resources more conservatively and thus appear suboptimal to nominal solutions, we believe that they will outperform nominal solutions in practice because they escape the constraint violations from which the nominal solutions suffer. While it is clear that this should affect at least the constraint violation metrics we posed, by our foregoing discussion on the dependency of objective value and constraint violation, it is possible to see that increased constraint violation will similarly impact the simulation values. Furthermore, the conservative protected solutions should perform more predictably in practice and have lower variances among these metrics.

The question then is to determine the factors that lead to increased violation of nominal solutions. The first obvious factor is the amount of uncertainty present in the system. If no uncertainty is present, it is unnecessary to protect solutions from uncertainty, and nominal solutions will be optimal. If much uncertainty is present, nominal solutions are likely to suffer from constraint violations and be outperformed by protected solutions.

In addition to this, problem characteristics play a role in the degree to which the uncertainty in the system affects constraint violation. Generally speaking, constraint

violation should increase as solutions increasingly utilize available resources. In terms of the formulation, this occurs when constraint slack approaches zero. This situation is prompted in different ways for different constraint types. As mentioned, the constraints subject to uncertainty in our formulation are of Type 1, 2 or 3 (see Table 5.1). They represent constraints with many binary variables that are all multiplied by coefficients that are subject to uncertainty (Type 3 constraints), and mixed constraints (binary plus continuous variables) where only the binary variables multiply uncertain coefficients (Type 1 and 2 constraints). Among the mixed constraints, Type 1 constraints contain 2 uncertain coefficients, and Type 2 variables contain only one uncertain coefficient.

For the purpose of this discussion, we divide each constraint into two parts: the part that is to the left of the equality (or inequality) we refer to as the Left Hand Side (LHS), and the part to the right we refer to as the Right Hand Side (RHS). The LHS typically includes the decision variables and their multiplying coefficients, and the RHS generally contains a constant that limits, or constrains, the decision variables. Slack, then, is eliminated when the LHS approximates the RHS in a solution. In MIP problems, this should happen as LHS coefficient values are decreased in relationship to the RHS (or, by stroke of luck, have coefficients that are near perfect factors of the RHS). In cases where the integer variables are binary, as with the ones in question, however, it is also necessary that there are a sufficient number of variables in the LHS, and that these variables are allowed to take on non-zero values. In the constraints that satisfy the former condition (i.e. Type 3 constraints in our model), the latter condition equates to allowing individual UAVs to visit more targets, and because of the one UAV per task rule, either increasing the number of targets or decreasing the number of UAVs. Additionally, any other constraints, outside of the Type 3 constraints, that limit the number of tasks a UAV can accomplish, should be relaxed (i.e., the constraint limiting the number of strikes per Type A UAV to two). In our model, then, it is expected that Type B UAVs are more prone to violating Type 3 constraints than are Type A UAVs. They experience travel times that are shorter than Type A UAVs, have shorter ISR service times than do Type A UAVs, and are

not limited in the number of tasks they can perform.

In the case of the mixed constraints where there are only a few variables, tightness should occur in the solutions when the continuous variables remove the slack between the constraint LHS and RHS. In both the Type 1 and 2 constraints, the continuous variables represent the time of arrival at a particular node. The formulation adds a small objective function weight to UAV return time, and earlier return times are given slight preference. Because of this, we expect that optimal solutions will assign values to the arrival times that remove slack and predispose the constraints to violation in the subsequent simulation.

Tightness in these constraints should also occur when target opportunity windows (or, more specifically, the intervals between the start and end of an opportunity window at a particular target, as well as the interval between the start of the window for one target and the end of the window for another target) decrease while the required service and travel time values increase. This limits the ranges within which feasible solutions can be found, and will also force optimal solutions to the boundaries of these ranges.

We now discuss the effect of uncertainty on constraint violation in greater detail. We assume symmetric uncertainty distributions that are centered about the mean values of the coefficients. We expect that constraint violation will increase as the ranges for the uncertainty increase, as the probability distributions of the uncertain coefficients contain more mass at the limits of their ranges, and as the number of uncertain variables in the equations increase.

It should be noted that, when symmetric mean-centered distributions are assumed for the uncertainty, the expected number of violations in a constraint should only increase with these three factors when the LHS is strictly less than the RHS in a solution. If, for a particular constraint and solution, the LHS is equal to the RHS, the expected number of constraint violations should not change with these factors, and should actually decrease if the LHS exceeds the RHS. Because solutions, by definition, are feasible to at least the mean coefficient values, the latter will never occur. This type of behavior can be depicted by plotting the distribution of the

uncertainty against a vertical line that indicates the RHS cut-off value. It occurs because all of the changes are symmetric about the mean, which we denote by $\bar{x}$; the effect on the particular constraint depends on the relationship of its LHS cut-off to the $\bar{x}$ value. Varying the distribution according to the first two factors (changing the magnitude of the hat value, denoted by $\hat{x}$, and altering the probability distribution) will change the percent of the probability density that occurs to the left (the feasible side) of the RHS limit, thus altering the probability of constraint violation. In the case of the third factor, the behavior can easily be illustrated by comparing feasible regions of single uncertain coefficients versus jointly varying coefficients. Because of the linearity of our constraints, the line of feasibility for a constraint with two uncertain variables is a derived distribution that cuts across equal values of the sum of the variables. For this third factor, we refer to the situation where already-existing variables change their status from being deterministic to being stochastic. In the case where previously nonexistent stochastic variables enter the expression, nothing can be said because the solution itself may change, but we anticipate the same outcome. Figure 5-6 demonstrates the change in the probability of violation among the three possible cases as $\hat{x}$ increases. Figure 5-7 depicts the change as the distribution of the uncertainty is "flattened" out, and Figure 5-8 depicts the effect of joining variables to the set of uncertain variables in the constraint. In Case 1 of each figure, it is assumed that the plan calls for a LHS that exceeds the RHS. Case 2 is the situation where the constraint is "binding" and achieves equality in the plan. Case 3 is an example of a situation where slack exists and the LHS is strictly less than the RHS. The lines that correspond to the three cases denote cut-off values for x for which the problem is feasible. Values for x that fall to the left of these lines are feasible, while those that fall to the right of the line cause the constraint to be violated. The probabilities of violation are solved using the geometry of the feasible regions about these lines, and are represented by the area of the feasible region that falls to the right of the lines.

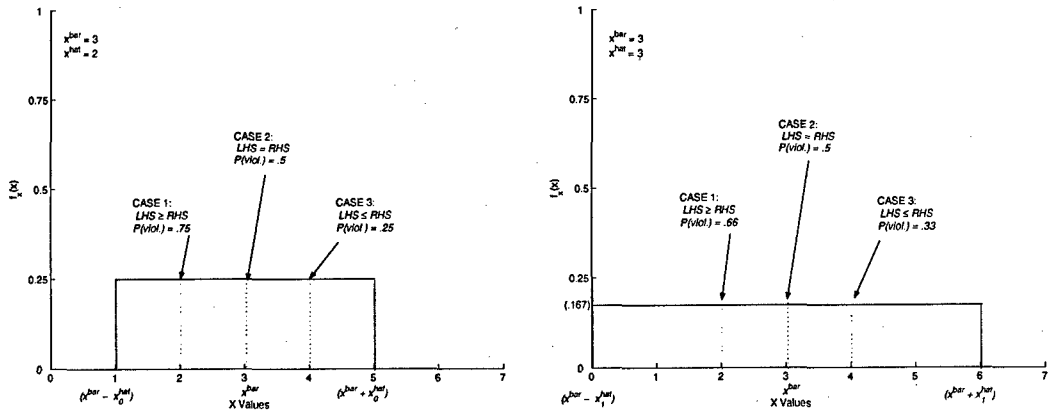


Figure 5-6: Increasing $\hat{x}$

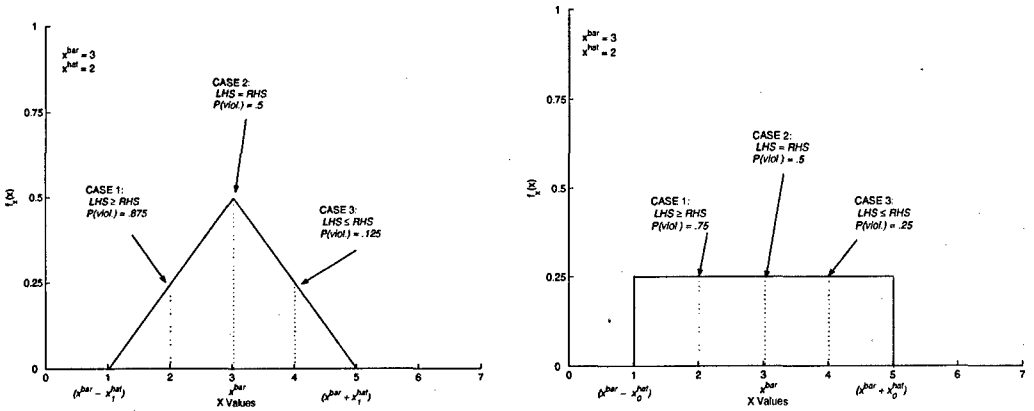


Figure 5-7: The effect of flattening the distribution of uncertainty

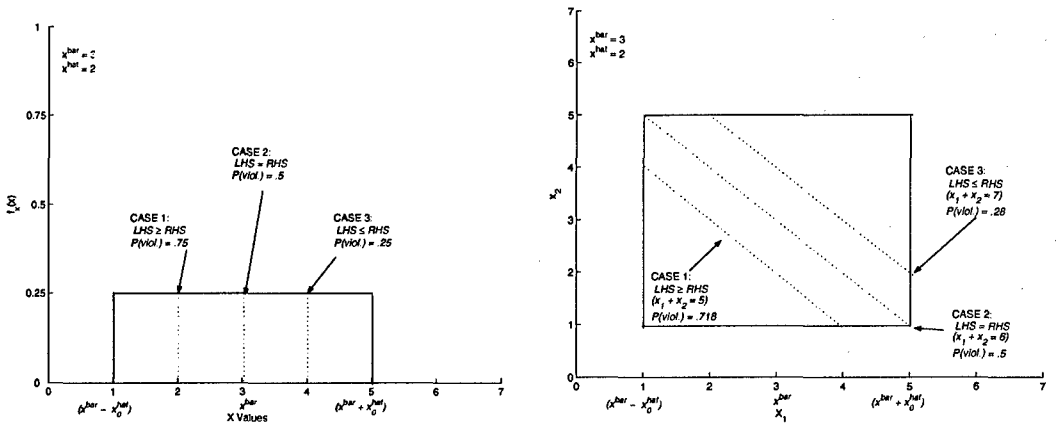


Figure 5-8: Increasing the number of uncertain variables

To summarize the discussion of this section, we believe that the performance of solutions with respect to the metrics improves with protection as constraint violation increases. Constraint violation increases as uncertainty increases, but also as the LHS of the constraints are allowed to approach their RHS limits in the optimal solutions. This belief guides the secondary hypotheses that are presented below.

Secondary Hypotheses

1. On average, protected solutions perform better when opportunity windows are smaller

2. On average, protected solutions perform better as opportunity windows overlap

3. On average, protected solutions perform better as service times increase

4. On average, protected solutions perform better as inter-target travel times increase

5. On average, protected solutions perform better as hat values increase

6. On average, protected solutions perform better as UAVs accomplish more missions

7. On average, protected solutions used in conjunction with a contingency planner outperform both unprotected solutions and solutions that are only protected

These hypotheses attempt to define the circumstances in which our Base Hypotheses hold and protection proves effective. Solutions "perform better" when the Base Hypotheses hold and when the metrics experience greater improvement with the same increases in protection. Hypothesis 1 is valid, for example, if the metrics experience greater improvements with protection for scenarios where opportunity windows are smaller than they do for scenarios with larger opportunity windows. The last hypothesis concerns the contingency planner in the simulation that was introduced in this section and will be fully developed in the next. We use these Secondary Hypotheses

as a guide in our experimentation, but also give ourselves the flexibility to explore interesting observations as they arise in the testing.

We now step through the experiment results. Each experiment is contained within its own section. The sections provide a motivation for the scenarios, present the hypotheses tested and scenarios used, present the output, and provide analysis for the experiment.

5.4 Tests and Results

5.4.1 Experiment 1

Motivation

Experiment 1 is intended to test the robust formulation and its solutions on what we feel to be a realistic platform. Instead of considering specific secondary hypotheses to test, we let the scenarios "drive" the test, and analyzed the results for an indication of how the protected solutions compared to the nominal ones. For this reason we spend more time developing the scenarios in this experiment, and try to place them in the context of a conflict as it unfolds.

Hypotheses

- Base Hypotheses

Scenarios Tested

Scenario 1 represents the initial stage of our conflict with the fictitious nation. Our forces have been prepositioned in an airbase belonging to a neighboring ally, and all missions to the targets dispersed within the enemy country must originate and return to this base. Because we are in the preliminary stages of the conflict, the main effort of the engagement is to obtain air superiority over the airspace above enemy territory. The primary task involved in securing air superiority is that of seeking out and destroying the enemy Surface-to-Air Missile (SAM) sites that are dispersed

throughout the country. At present, such tasks are accomplished by "Wild Weasel" squadrons, which fly directly into the radar ranges of these SAM sites in order to provoke the site operators to lock in on their aircraft. This allows the pilots to locate the SAM sites and target them. The job is dangerous and stressful, and perfectly fits the criteria of a job that might be given to the UAV unit of the future. The figures below provide a layout of the targets, as well as an overview of the data inputs used for the scenario.

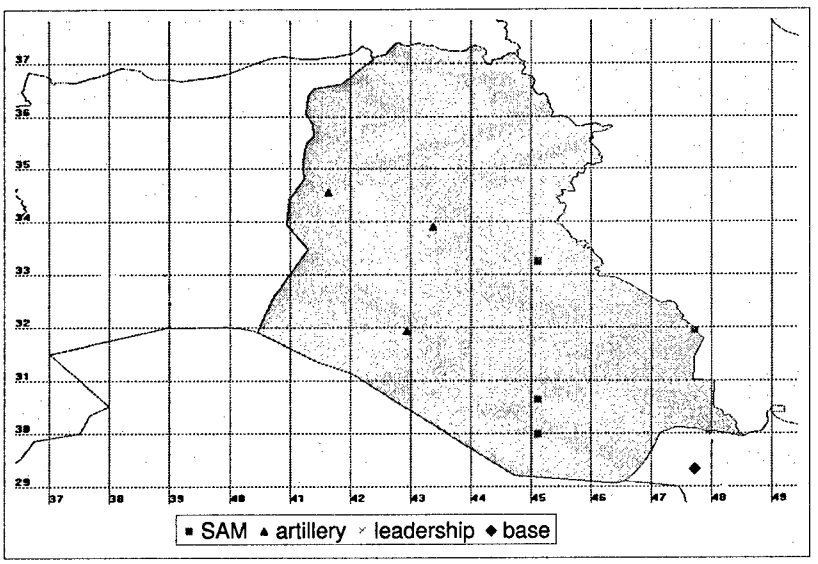


Figure 5-9: Map of Scenario 1

Scenario 2 represents a later stage in the same conflict. Friendly forces have infiltrated the enemy country and are now seeking to extinguish a few remaining pockets of resistance. Ground forces have secured a strategic location in the heart of the country, and a friendly airbase is now fully operational at this location. The UAV unit has been repositioned to and now operates out of this location. Targets now consist primarily of Mobile Artillery Units (MAUs) and suspected leadership elements that are centered at the remaining pockets of resistance.

Scenario 3 simulates a situation where either the existing intelligence concerning the enemy is poor, or the enemy or environment are highly unpredictable. Our information on target type and location is poor, and as a consequence, expected task

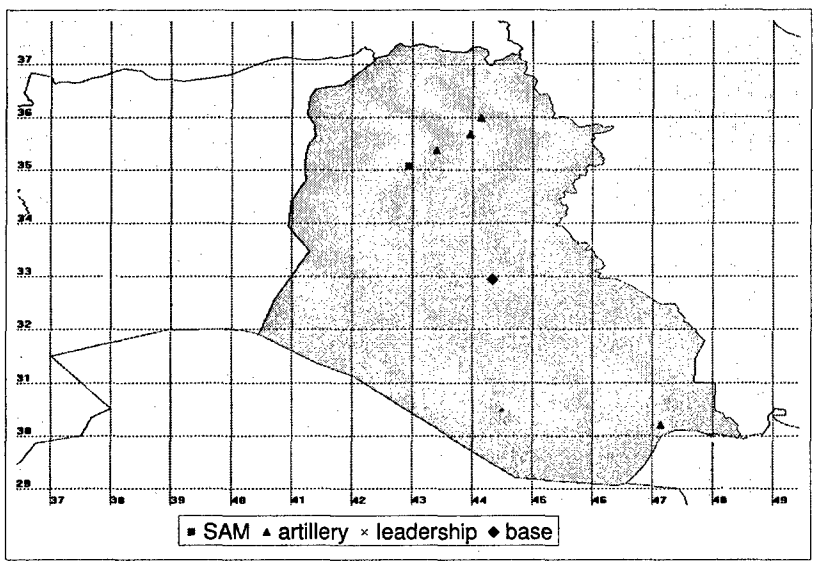


Figure 5-10: Map of Scenario 2

rewards, service times, and target opportunity windows are larger than before and have greater levels of uncertainty surrounding them.

Scenario 4 is a variation of Scenario 1 that simulates operations on a day with poor weather. Expected travel and service times are longer than before, and uncertainty in all of the stochastic values is likewise greater than before. This scenario differs from Scenario 3 in that now inter-target travel times also take on greater values and have a greater level of uncertainty surrounding them.

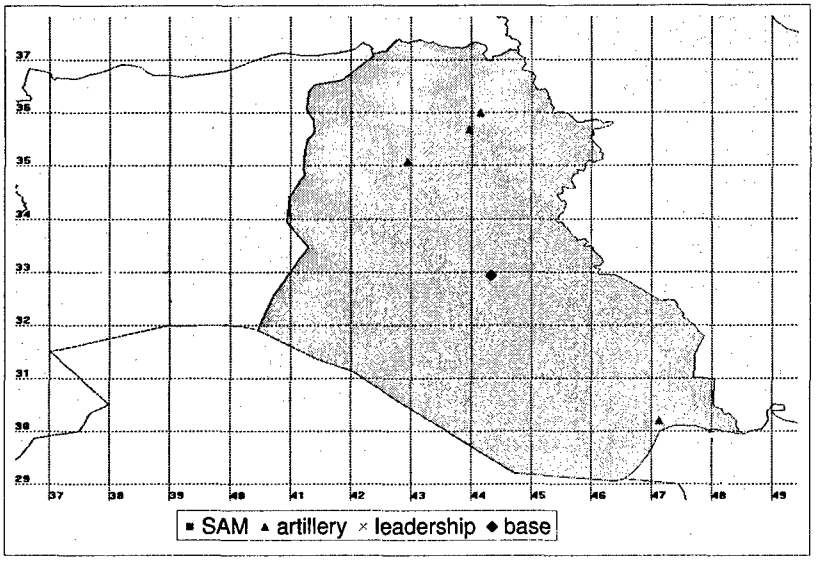


Figure 5-11: Map of Scenario 3

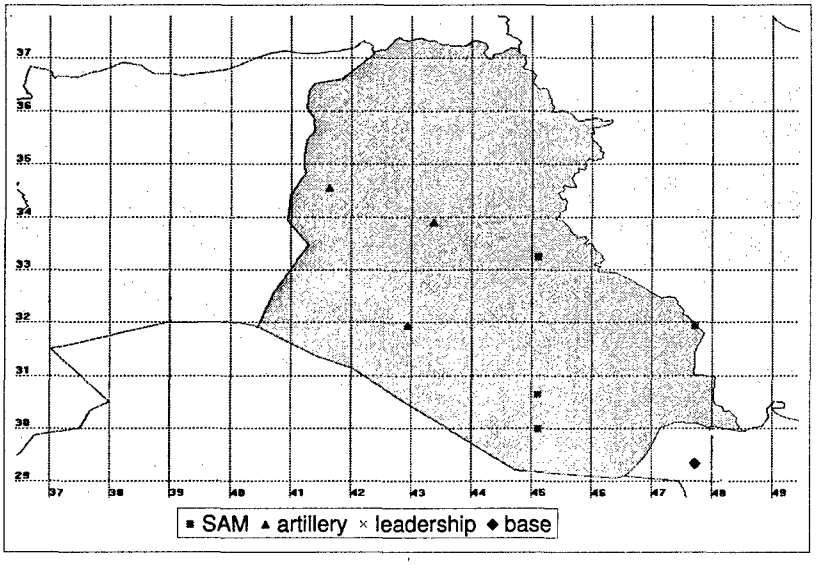


Figure 5-12: Map of Scenario 4

Results and Analysis

We now step through the results and analysis for each of the scenarios.

Scenario 1 The first observation concerning Scenario 1 is that, against our expectations, average simulation value decreases as protection level increases. The decrease is, for the most part, a two-step monotonic decrease in the Nominal Sim, but actually increases before decreasing in the Smart Sim. The Nominal and Smart Sims mirror one another, but the Nominal Sim, unexpectedly, outperformed the Smart Sim in simulation value at 0% protection. Both value curves appear to exhibit a slight waviness in the protection range from 20% to 80%. The reported objective function value also decreases monotonically, but drops much more dramatically with protection.

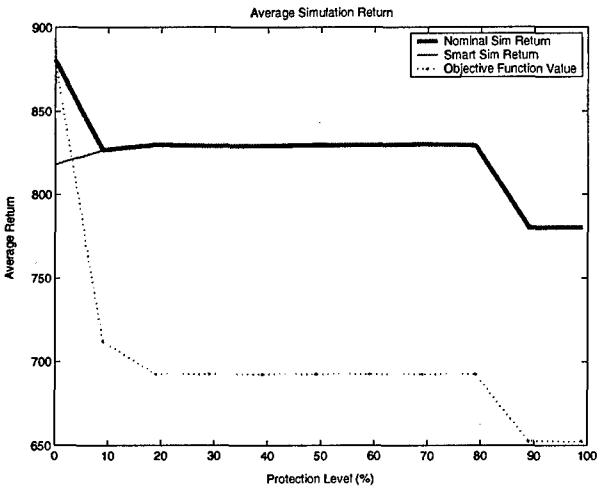


Figure 5-13: Sim and objective function values for Experiment 1, Scenario 1

As predicted, the average value for the number of violations does diminish with protection. The decrease is evident among all violation types, although very few Type 2 or 3 violations are reported, and all Type 2 and 3 violations disappear completely by 30% protection. Once again, the Nominal and Smart Sims are essentially identical, and differ only marginally at 0% protection. The Smart Sim results report more Type 2 violations at 0% protection than do the Nominal Sim, but Type 3 violations in the Smart Sim slightly undercut those of the Nominal Sim. The "lateness" statistic,

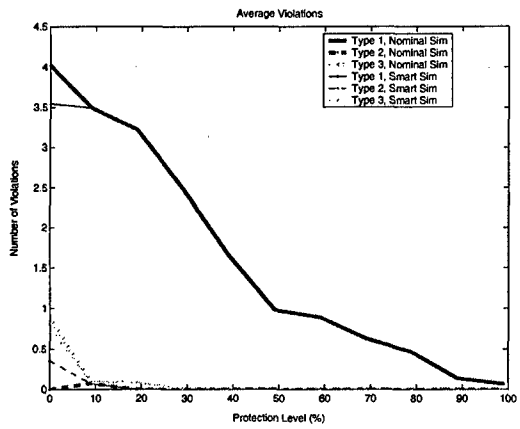
which is a similar measure and reports the amount by which Type 1 violations exceed their RHS bounds, performs closely to the reported Type 1 violations, but tends to be smaller than them. The slope of the two curves also differs, with the lateness curve diminishing more quickly.

The earliest times that these constraint violations occur, which we refer to as the earliest "departure" times, occur progressively later for Type 1 violations. While the trend is also towards later violations among the Type 1 and 2 violations, the curves of these two metrics experience two dramatic dips at the 10% and 70% level, respectively.

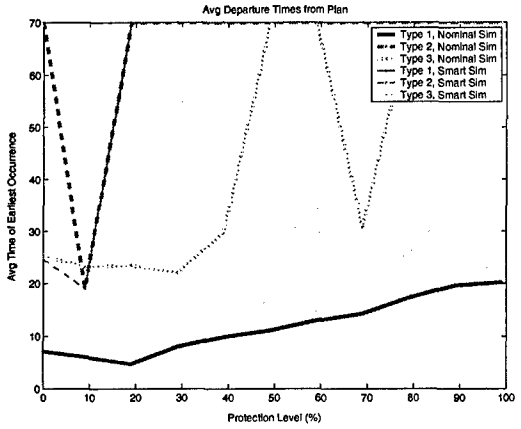
The ratio of the simulation value to objective function value perform contrary to what was predicted, and increases with protection level. This is depicted by the gap between the simulation and objective function values in Figure 5-13. The general trend among the variances of the foregoing metrics over the protection levels is that of a decrease. The drop is dramatic for the Smart Sim at the 10% level, while the Nominal Sim remains relatively constant throughout all of the protection levels. The decrease in the variance of lateness is also apparent in both simulations.

We now provide an analysis of the results. The fact that simulation value decreases with protection is understandable when viewed alongside the constraint violations (Graph (a) of Figure (5-14). Our prediction that expected simulation value would increase with protection level was based on the expectation that the protected solutions would experience fewer constraint violations. This behavior for simulation value can only be observed when Type 2 and Type 3 violations occur, however, as only these violations have potential to directly impact the simulation values. As mentioned earlier, however, almost no Type 2 and 3 violations occur, but rather, the vast majority of the violations are Type 1 violations.

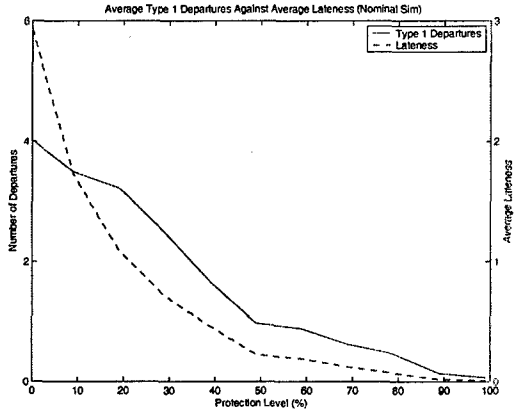
We feel that objective function value drops so dramatically due to the way that the objective function coefficients are protected. Expected values for protected coefficients are based on their worst-case values ($\bar{C} - \hat{C}$). As a result, objective function values tend to be overly pessimistic, and the simulations post values that are considerably higher. The monotonic objective function value decrease is expected. Adding



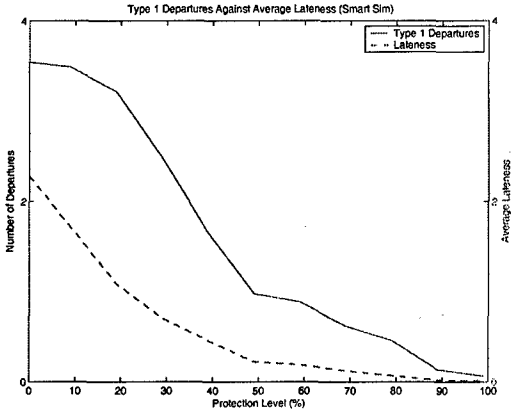
(a) Constraint violations



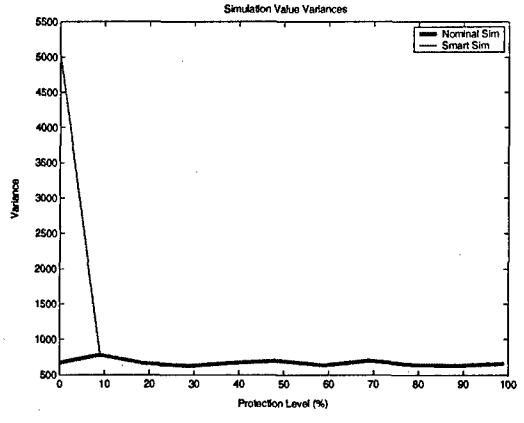
(b) Departure times



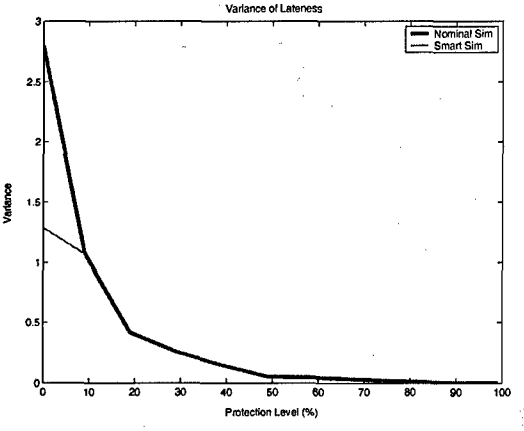
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-14: Experiment 1, Scenario 1 results

protection acts to shrink the problem feasible region. Because the feasible regions at higher protection levels are subsets of those at lower levels, optimal solutions at these levels must necessarily be less than or equal to those at lower protection levels. The fact that changes in the objective function curve are not necessarily accompanied by changes in the average simulation values is also understandable, as these drops might be due entirely to objective function hat values. Such changes are in contrast to ones where the new plans perform a set of tasks that bear a lower expected objective function value. If the sums of $\bar{C}x$ are identical in two plans, the simulation values of the plans would presumably also be similar, even if their reported objective function values differ.

The increase in value experienced by the Smart Sim over the 0% to 10% protection range can be attributed to the Type 2 violations documented in that range. The main distinction between the two simulations is that the Smart Simulation anticipates Type 2 and 3 departures and attempts to recover from them. When the Smart Sim anticipates a Type 2 violation, it skips to the next target in order to conserve UAV range for future targets. This maneuver still prompts a Type 2 violation, but presumably improves the likelihood that future targets will be successfully visited. When the simulation anticipates a Type 3 violation, it skips the target in question and attempts a return to base. Once again, such an action prompts a Type 2 violation, but hopefully averts the more serious Type 3 violation. Therefore, while the Smart Sim does perform marginally better in that it has fewer Type 3 violations, it incurs more Type 2 violations in the process. These violations account for the disparity between the Nominal and Smart Sim values at the 0% level. While the Smart Sim's decision to skip a target might successfuly avert a Type 3 violation, it might also divert a UAV from a target that it could have successfully prosecuted, as is evident by the Nominal Sim's outperformance of the Smart Sim.

The high degree of similarity between the two simulations apart from this one point is logical. As mentioned, the two diverge mainly in their handling of possible Type 2 and 3 errors. Because almost none are reported throughout the testing (especially after the first few protection levels), the two plans are essentially identical.

While the trend of the average departure time generally follows our prediction, the two drops among the Type 2 and 3 violations somewhat mask this trend. As is evident in the figures, while the results of the departure times are constistent with those of the average violation counts, they tend to be more dramatic. This is likely due to the way in which we compute the departure time statistic, as we now describe. Rather than averaging over all iterations in order to compute the average earliest departure time, only iterations with nonzero violation counts are considered. This is because iterations with no violations of a certain type are recorded as having a departure time of 70 hours for that constraint type (a time after the required return time for all vehicles). Averaging over all iterations, then, would allow iterations with no violations to have an undue influence on departure times. A consequence of handling the statistic in this way, however, is that occurrences which are not statistically significant can still have a large impact on the metric. In particular, the change in departure time at 10% protection is caused by only two iterations, out of the 1000, posting Type 2 and 3 violations. This metric was designed to be viewed alongside the average constraint violation count, so that a more complete picture of the constraint violations might be gained. If no violations of a particular type are discernable on the graph of average constraint violations, corresponding departure time values that are lower than 70 should not weigh heavily on conclusions about solution performance.

The fact that the value of lateness tends to be less than the number of Type 1 violations reveals that many of the Type 1 violations exceed their RHS limitations by little (by less than 1 hr). In the graphs of lateness for the Nominal and Smart Sims it is apparent that, while Type 1 violations are similar between the two simulations, the Smart Sim posts values for average lateness that are substantially smaller. We consider lateness only in Type 1 violations, as we deem that this statistic is meaningless for the other violations. As in the discussion of the metrics section, Type 2 and 3 violations are irrecoverable, and the amount by which they are violated is therefore irrelevant. In the case of Type 1 violations, however, we feel that the amount by which constraints are violated does impact solution quality, and we thus record the metric.

The fact that Smart Sim simulation value variance is so high at 0% protection is likely due to the occurrence of few Type 3 violations at this protection level. Because these violations disappear at higher protection levels, the variance quickly drops off.

Scenario 2: The results for Scenario 2 are generally consistent with those of Scenario 1, but a few differences are noticeable. First, the graph of the objective value indicates dips at the 70% and 90% protection levels. These two dips are reflected in the simulation values of both simulations, which remain constant throughout the remainder of the protection levels. The graph of the constraint violations is also similar to Scenario 1, exhibiting drops among all violation types, but with fewer Type 3 violations than Scenario 1. The graph of constraint violations shows a slight increase in the occurrence of Type 2 and 3 violations in the range from 0% to 10% protection, which then drops off by 20%. The Smart Sim posts more Type 2 violations at the 0% level than does the Nominal Sim, but Type 2 violations for the Nominal Sim then rise to the Smart Sim level by 10% protection. The graph of the earliest departure times indicates a general trend of later departure times but reports a pair of spikes in the departure times for Type 2 and 3 violations. As in Scenario 1, the value for lateness is less than the number of Type 1 violations for both simulations. Variance for lateness declines steadily, but the variance among the simulation values spikes at 10% before declining as well.

The dips in the objective function value are unexpected, and indicate that the solver was not able to find the optimal solution. Runtime results, however, indicate that the solver terminated successfully on all protection levels and was able to find a solution within the specified 5% of optimality within the 5000 second time limit (Figure 5-17). Figure 5-17 also reports the optimality gaps for the tested protection levels, and reveals two corresponding spikes in optimality gap at the 70% and 90% levels. While the solutions at the 70% and 90% levels fall within the required optimality gap, they are not optimal in the sense that solutions with greater objective function value (and smaller corresponding optimality gap) exist. This is evident because a solution with the higher objective function value is feasible at the more restrictive 100% protection level. The spikes would not have occurred, had the required optimality

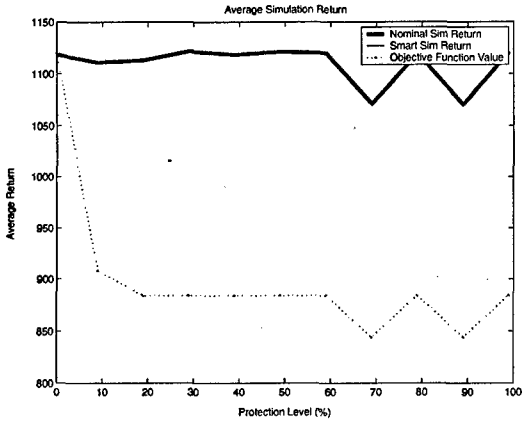
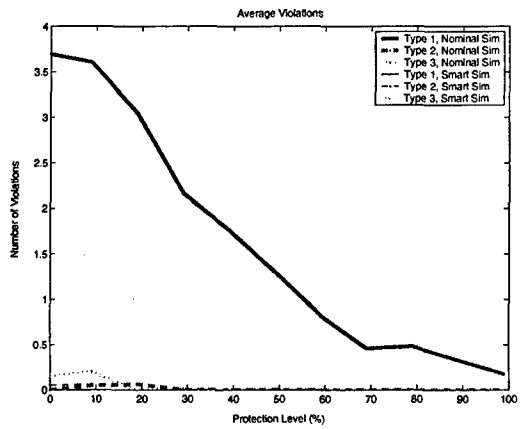


Figure 5-15: Sim and objective function values for Scenario 2

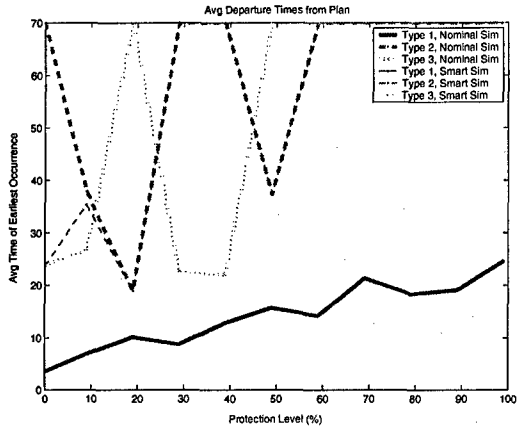
gap been smaller. It should be noted that a solution with nonzero optimality gap is not necessarily suboptimal, and that such a gap merely indicates the potential for a better solution. Optimality gap is calculated through the use of a bound, and not by the value of a known feasible solution. In practice, the branch-and-bound algorithm may find the optimal solution early in its search, but continue to search because the required optimality gap has not yet been achieved. What ensues is an often lengthy process of pruning that finally causes the bound to decrease to within the required optimality gap of the found solution. It is for this reason that we decide to maintain the 5% optimality gap.

The fact that the simulation values for both simulations remains constant outside of these two dips indicates that no simulation value must be given up in order to attain a higher level of protection. As in Scenario 1, close to no Type 2 and 3 violations occur throughout the testing, and the added protection only helps to reduce Type 1 violations.

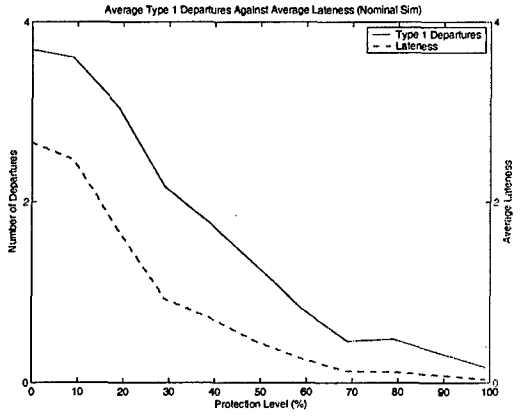
It is interesting, however, that Type 2 violations actually increase in the protection level range from 0% to 20%, and that Type 3 violations increase through the 10% level, before both drop off. The vacillation observed in the reported departure times for Type 2 and 3 violations in the first few protection levels is similarly unexpected. Because Type 2 and 3 violations have the potential to impact simulation value, the behavior of the violations within the first few protection levels has an impact on



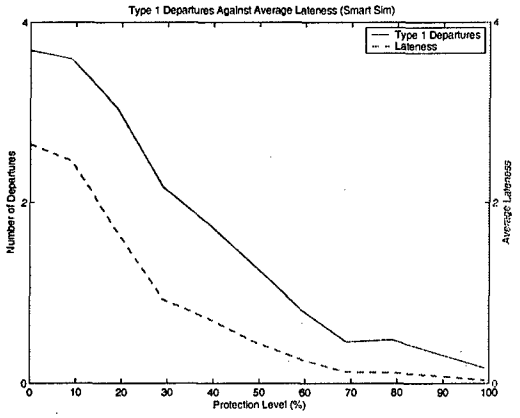
(a) Scenario 2 constraint violations



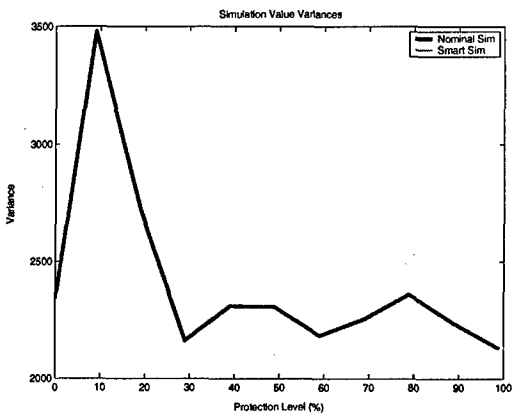
(b) Earliest departure times for Scenario 2



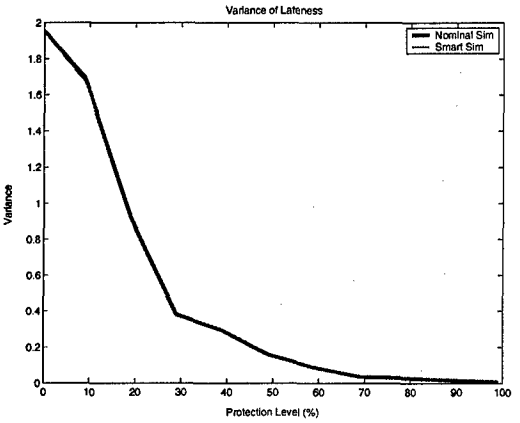
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-16: Experiment 1, Scenario 2 results

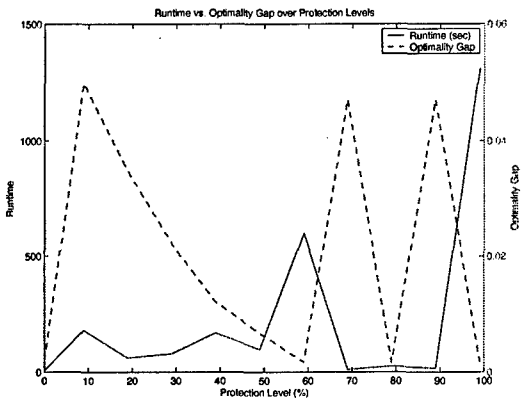


Figure 5-17: Runtime and optimality gap for Scenario 2

simulation value variance, as is confirmed by the spike in this variance at the 10% level. These results indicate that improvement in these metrics is not guaranteed for solutions within these first few protection levels.

Scenario 3: Results for Scenario 3 are similar to the previous two in that the general trend for constraint violations is downward, the trend of earliest departure times for Type 1 violations is for later departure times, lateness remains below the Type 1 constraint violation count for both simulations, and the variance for lateness decreases with protection outside of a spike at the 10% level. In Scenario 3, the vacillation in Type 2 and 3 violations extends until 40% protection, and occurrences of Type 2 and 3 violations appear as late as the 80% and 70% protection levels, respectively. As in Scenario 2, the graph of earliest departure times reveals the erratic behavior in the earliest departure times of Type 2 and 3 violations. Scenario 3 results are different from those in the previous scenarios in that the reported simulation values increase above the 0% protection objective function value before decreasing again.

The fact that the simulation values increase briefly in the protection range from 10% to 40%, and that this increase puts them above the unprotected objective function value is unexpected. Upon further inspection, we find that the solution to the 0% protection level chooses not to perform a strike at Target 7, while the plans in the 10% to 40% protection range do. This target is the Mobile Artillery Unit (MAU)

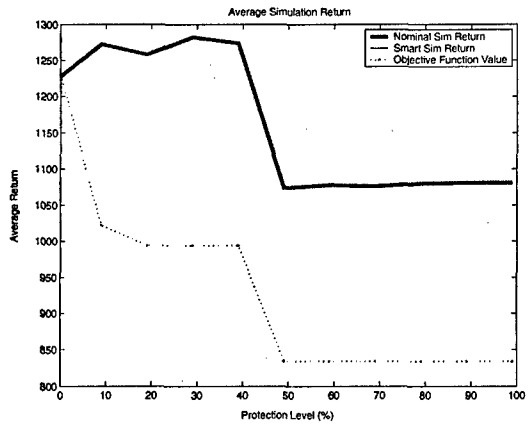
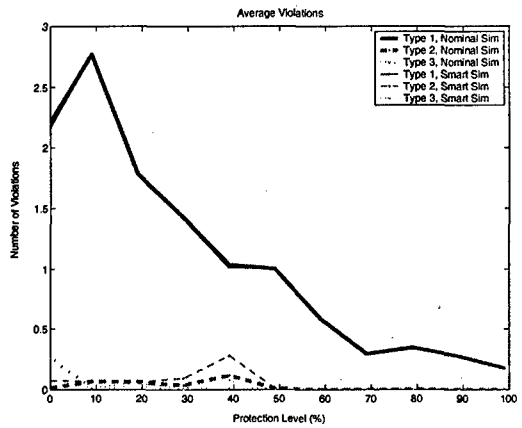
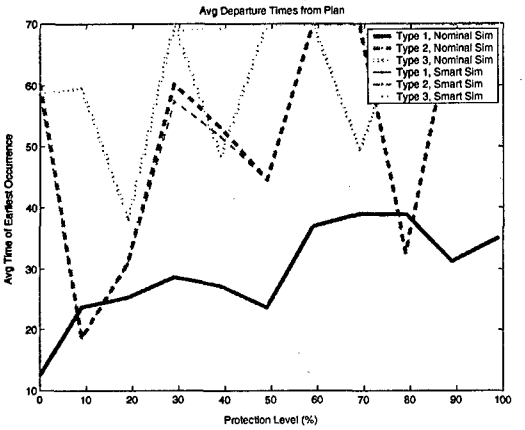


Figure 5-18: Sim and objective function values for Scenario 3

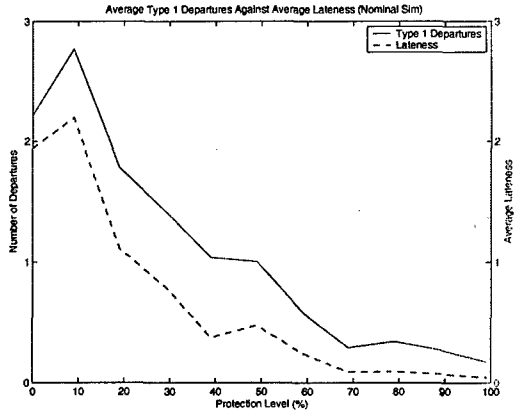
at the bottom right corner of the map, and the value associated with striking it is apparently outweighed by the objective function travel costs incurred from visiting it. Because of the routing that the protected plans choose for their UAVs, this target becomes worthwhile, and they choose to strike it. As a result, the protected solutions appear to return simulation values that are higher than the unprotected objective function value. The marginal travel costs ($\frac{1}{100}$ of the travel time) were included in the objective function in order to ensure that routes reflected the shortest path through the targets. Because it was not our intention for these costs to impact choices to visit targets, we choose to scale them back in future runs.



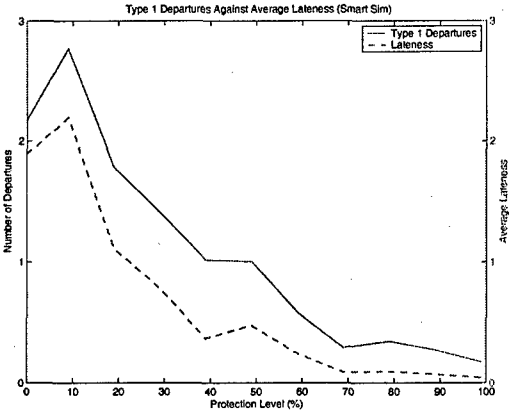
(a) Scenario 3 constraint violations



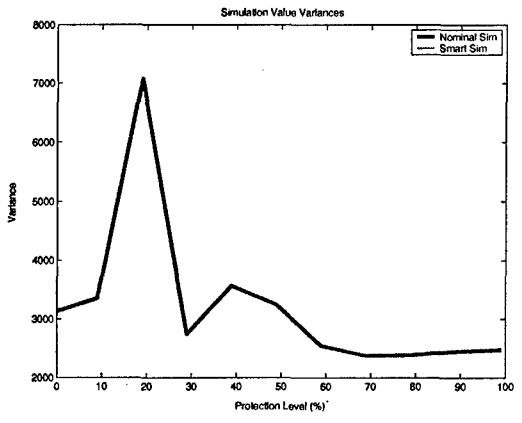
(b) Earliest departure times for scenario 3



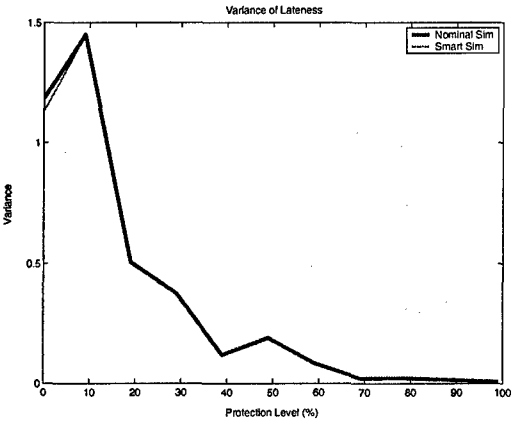
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-19: Experiment 1, Scenario 3 results

Scenario 4: Simulation values in Scenario 4 are similar to those of Scenario 1, in that the Smart Sim value is significantly lower than the Nominal Sim value at the 0% protection level, but rises to meet Nominal Sim value at the later protection levels. The separation between the results of both simulations is more pronounced, and the two simulation values do not join until the 20% protection level. Outside of this, the results for constraint violation, earliest departure time, and variance are more consistent in Scenario 4 than in the previous scenarios, and contain fewer spikes and oscillations.

A likely cause for the great deal of separation between the values of the two simulations is the significant increase in Type 2 violations reported for the Smart Sim over the Nominal Sim. These violations occur relatively early in the simulation, and thus have the potential of affecting simulation value. While the Nominal Sim does not suffer from Type 2 violations, it experiences more Type 3 violations in both the 10% and 20% protection levels. Type 3 violations occur at a greater frequency in Scenario 4 than in the previous scenarios. The reported variance is much more consistent for Scenario 4 than it is for Scenario 3; variance for the Nominal Sim remains constant throughout the testing while the variance of the Smart Sim simulation value is high at the 0% protection level but then drops to join the Nominal Sim level. The higher average for Type 2 violations in the Smart Sim again points to the simulation skipping targets that it could have attended.

We conclude with a few general comments on Experiment 1. In general, the results did agree with our hypotheses concerning lower constraint violations and lateness, later plan departures, and lower variance (Base Hypotheses 1, 2, 3, and 6). We found Base Hypothesis 5 regarding the ratio of achieved value to objective function value to be wrong, gaining the intuition that protected objective function values were overly pessimistic and that the realized objective function coefficients were typically higher. Base Hypothesis 4 was similarly found to be incorrect. Realized values tended to decrease with protection, and unprotected solutions appeared superior in terms of the simulation value metric. Unprotected solutions were unimpeded by Type 2 or 3 constraint violations, and therefore were able to achieve all of the tasks

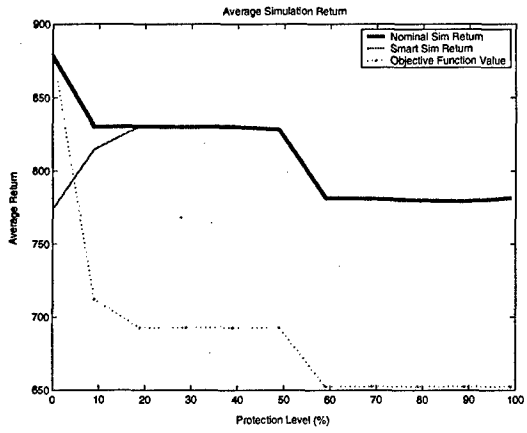


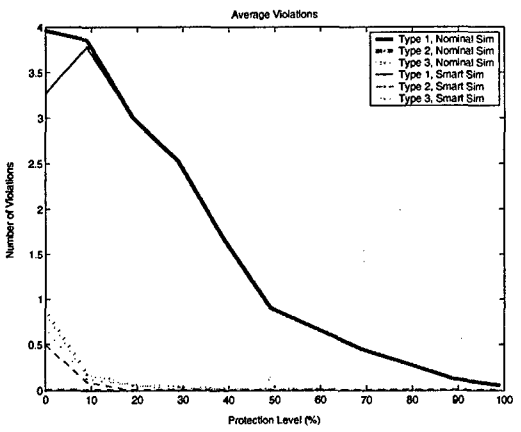
Figure 5-20: Sim and objective function values for scenario 4

that they planned for. This matter highlighted the fact that uncertainty did not heavily influence the simulation outcome. In fact, added protection had the effect of increasing Type 2 and 3 violations and the variance of the simulation value in the earlier protection levels of some scenarios. As noted earlier, solutions with added protection did not guarantee better performance in these metrics until the protection level increased to a certain point.

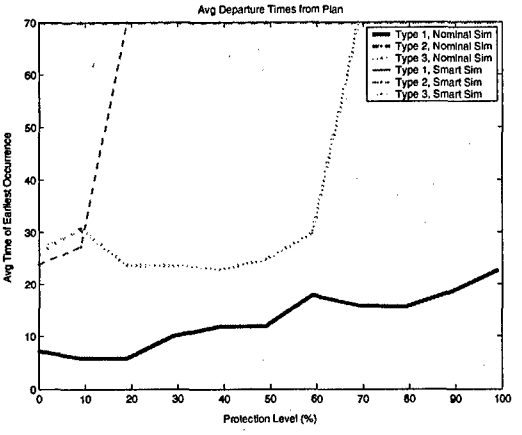
In addition, it was apparent from the simulation results, that while there were slightly fewer Type 3 violations in the Smart Sim than in the Nominal Sim, this decrease often came at the cost of lower value, higher value variance, and more Type 2 violations. The added precautions put in place in the Smart Sim seemed to be counterproductive, as they prevented the UAVs from visiting targets that could have successfully been prosecuted.

An observation related to the minimal influence of uncertainty is that increasing protection often did not greatly diminish the value that the plans were able to achieve. The Separation constraints, which do not factor into value received (a loss in simulation value does not necessarily accompany Separation Constraint violations), tended to be the only binding constraints, and adding protection to the solutions primarily impacted these constraints.

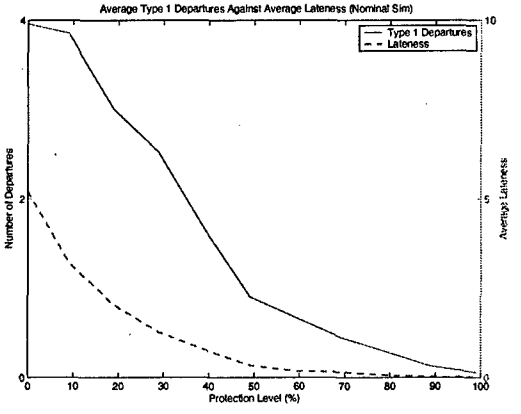
These two factors, that uncertainty did not appear to greatly impact plans, and that no tradeoff was required for added protection, both suggest that a "recalibra-



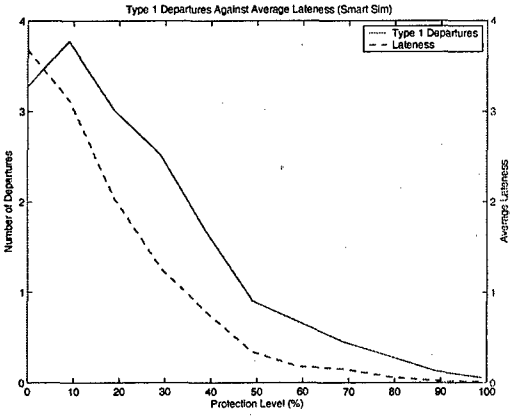
(a) Scenario 4 constraint violations



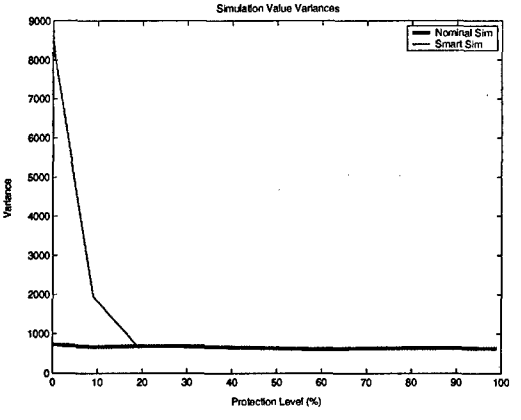
(b) Earliest departure times for scenario 4



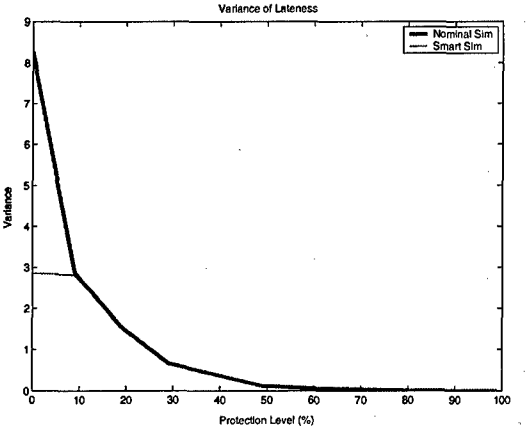
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-21: Experiment 1, Scenario 4 results

tion" of the scenarios might be appropriate. The entire premise for our considering uncertainty in our formulation is our belief that uncertainty can have a large impact on operations. We feel that this is true in practice, and thus feel that our data inputs should reflect this. In addition, because we desire to investigate the tradeoff between optimality and robustness, our scenarios and data inputs should cause constraints to be more binding, and require that such a tradeoff be made. This motivates our "recalibration" of the Base Scenario we use, which we discuss in the next section.

5.4.2 Calibration

In response to the observations from the previous experiment, we decide to implement some changes on the Base Scenario that we will use for the remainder of the experimentation. As mentioned in the Experiment 1 analysis, we observed that uncertainty did not seem to prevent the unprotected solutions from attaining all the planned value. The plans did not experience much in the way of Type 2 or 3 violations, and when they did, these violations occurred late enough in the simulation to not impact simulation value (Type 2 or 3 violations occurring on the return to the airbase do not affect simulation value). Furthermore, protected solutions were not required to give up much in the way of targets visited, and could often produce expected returns that approximated the nominal objective function value. The main rationale for introducing uncertainty into the solution process is the belief that uncertainty does impact operations. In order for us to capture the performance of this formulation when such is the case, we felt it necessary to increase the uncertainty level we used in the testing. Increasing uncertainty would also allow us to analyze any tradeoffs that might exist between protection and value. We decide to "recalibrate" the scenarios so that the effects of uncertainty on plans will be more apparent, and will afford us a testing environment where the tradeoffs for protection can be perceived. In doing so, however, we are constrained by our desire to maintain a realistic testing environment. While it is expected that uncertainty will play a greater role in the results as we increase the *hat* values, doing so might come at the cost of simulation realism. One would hope, for instance, that the hat values would be smaller in magnitude than

their corresponding mean values ($\bar{a}$). For this reason, we settle upon a Base Scenario where the majority of the *hat* values are 30% of the mean values. For the opportunity windows, this means that all but two of the targets are assumed to be "leadership" targets (the remaining two are MAUs), whose *hat* values comprise 30% of the length of the opportunity window ($\hat{A} + \hat{B} = .3(B - A)$). In addition to these changes, we decide to remove one Type A UAV and one Type B UAV. This change is an effort to further restrict the problem and to prevent the solutions from easily performing all tasks. We feel that doing so might allow us to perceive more of a tradeoff with protected solutions. From this point we continue the testing using the calibrated scenario as a baseline.

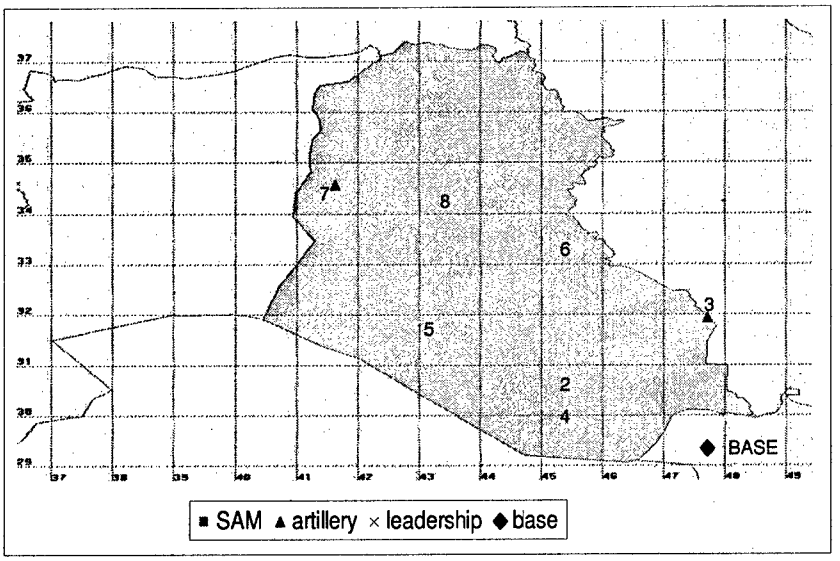


Figure 5-22: Map of Base Scenario

5.4.3 Experiment 2

Motivation

From Experiment 2 onward, our goal is to test the secondary hypotheses listed in the hypotheses section. Namely, using the new calibrated Base Scenario, we desire to

test when, and under which circumstances, the base hypotheses regarding protected-solution performance hold. Experiment 2 focuses on the impact of opportunity windows on the simulation results. Throughout the experiment, we test different target opportunity window configurations, while leaving other problem values constant. Our expectation is that protected solutions will perform better as opportunity windows are tightened.

Hypotheses

- On average, protected solutions perform better when opportunity windows are smaller.
- On average, protected solutions perform better as opportunity windows overlap.

Scenarios Tested

The two factors we perturb in the following scenarios are the length of the opportunity windows and their separation. The length refers to the time between their start and finish, while time window separation refers to the time between the bounds of opportunity windows for two different targets.

1. Base opportunity windows (Base Scenario)
2. Opportunity windows relaxed completely
3. Tight opportunity windows
4. Much opportunity window separation
5. Complete overlap of opportunity windows

Results and Analysis

Scenario 1 (Base Scenario): We first discuss the results of the Base Scenario that serves as a baseline for our perturbations. Observations from the results are that simulation value still goes down for the Nominal Sim as protection level increases,

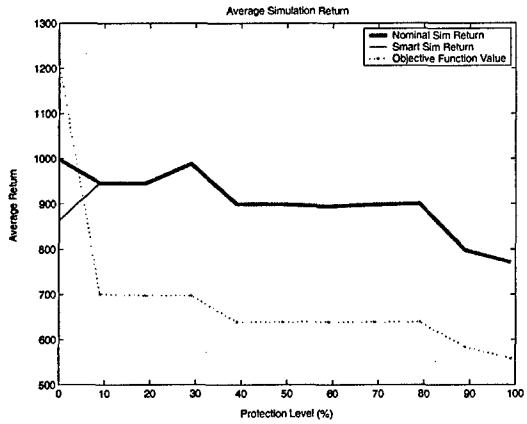


Figure 5-23: Sim and objective function values for Scenario 1

as it did for the previous experiment. As was often the case in the simulation value output of Experiment 1, the Nominal Sim outperforms the Smart Sim on value in the unprotected solution, and Smart Sim value actually appears to increase over the 0% to 10% protection interval. Values for both simulations join up after the 10% level. Both experience an increase in value at the 30% protection level and dips at the 40% and 90% protection levels, where corresponding dips occur in the objective function value of the solutions. An increase in objective function value is not reported at the 30% level. The objective function value experiences a large drop at the 10% protection level and two smaller drops at the 40% and 90% levels.

As in the previous experiment, the large drop in the objective function value reported at the 10% protection level appears to be due to the fact that two fewer tasks are performed at the 10% level, as reported by Table 5.4.3. The table is an itinerary of targets that are included in the plans at several protection levels. Nodes 1 and 9 both correspond to the airbase, and the other nodes correspond with the seven targets in the scenario (refer to Figure 5-22 for the target locations and types). An "I" in the space for a target indicates that the plan calls for an ISR of the target, while an "S" indicates a call for a strike. As Figure 5-23 reveals, the corresponding drop in simulation value at this protection level is much smaller. This appears to be due to the fact that most iterations at the 0% level are incapable of accomplishing all of the tasks in the unprotected plan (causing the unprotected simulation value

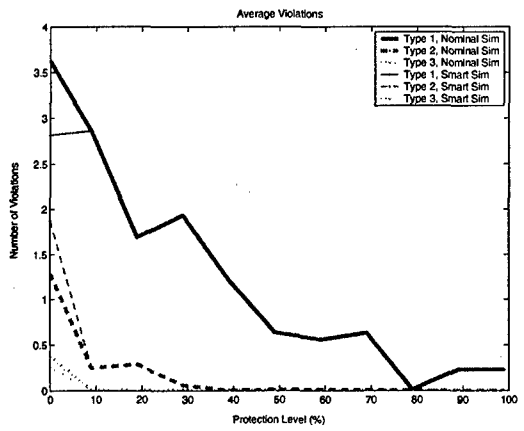
to be lower than the unprotected objective function value), and that a good deal of the drop in objective function value is due to an overly pessimistic expectation of the value of task accomplishment (causing the simulation value to be higher than the objective function value at the 10% level). Another interesting change in the plan at this point is the transition of all ISR responsibilities to the Type B UAV, and the relegation of all Type A UAVs to performing only 1 strike apiece. The two other drops in objective function and simulation value are due to swapping the targets that were visited, and both swaps replace a risky-but-profitable leadership target with a less risky and less profitable MAU (reference Table 5.4.3 and Figure 5-22). As noted earlier, while the reported objective function values for the 20% and 30% protection levels are the same, average simulation value actually increases over this interval. This increase corresponds to the decrease in the average number of Type 2 violations reported at the 30% protection level (see Graph (a) of 5-24). This drop is due to the fact that the arrival time at Target 6 is pushed back to 7.916 hours from 9.7 hours, giving the 30% plan more clearance from the close of the target's opportunity window (which occurs at hour 10). This change in planned arrival time leaves less buffer for the trip that the servicing UAV must make from the airbase to Target 6, and contributes to the increase in Type 1 violations within this range.

Table 5.4.3 also reveals that, as planned, this scenario is more constraining, as the unprotected solution calls for UAVs to perform only one task at three of the nodes and completely miss one of the nodes. We point out again, that while only the protection levels listed in the table report significant movements in the objective function value, this does not mean that the solutions do not change in other protection level ranges. While the reported objective function value remains the same at these ranges, the solution, in terms of target-to-UAV assignments and arrival times, is different, and this is reflected in the changes in the other metrics over these ranges.

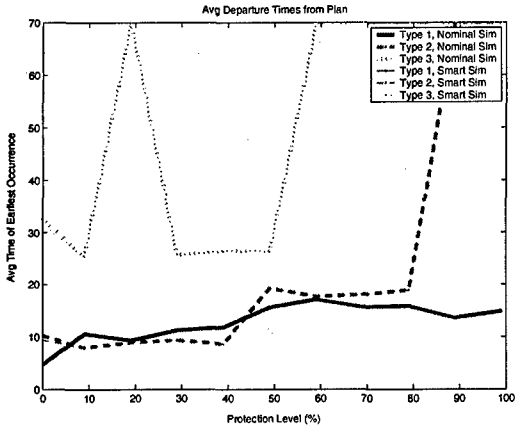
Graphs of Scenario 1 constraint violations and departure times appear to be very consistent. Outside of the already accounted for increase in Type 1 violations at the 30% protection level and another increase in the 80% to 90% level, constraint violations decrease monotonically with protection. Graph (b), which reports the average

earliest time of departure, indicates that, while the trend is for departures to occur later during the plan, the change in departure times is not monotonic over the protection levels. The line representing Type 3 violations is the most pronounced in this regard, and the time of the earliest Type 3 violation decreases at the 10% protection level. The spike at the 20% level indicates that the plan at this protection level incurs no violations, and the subsequent return to the 10% level reveals that, while the increase is very slight, the line for Type 3 violations in Graph (a) is also nonmonotonic. Relative to the previous scenarios, Type 2 violations persist longer into the protection levels, gradually occurring later and later, until they finally disappear at protection level 90%.

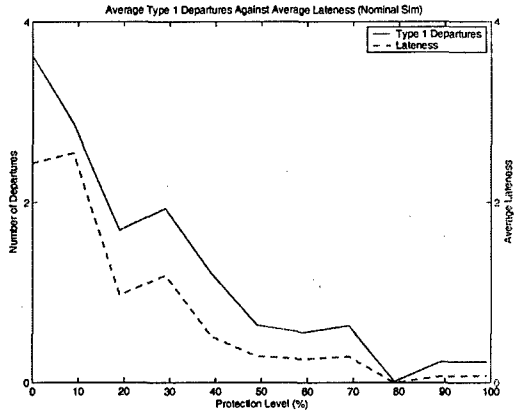
As was the case in Experiment 1, Graphs (c) and (d) report that average lateness shadows but remains below the Type 1 violation count. An interesting departure between the two lines occurs in the interval from 0% to 10%, where lateness actually increases during a time when the average number of violations decreases (in the Nominal Sim). This indicates that while the number of violations decreases, the amount by which each violation exceeds the RHS limit increases on average. Graph (e) of Figure 5-24 indicates that variance decreases quickly in simulation value, though it increases briefly at the 20% level. While Graph (f) also reports a drop in the variance of lateness, this drop is preceded by a sharp increase at the 10% protection level. The increase in variance corresponds with the increase in average lateness over this interval.



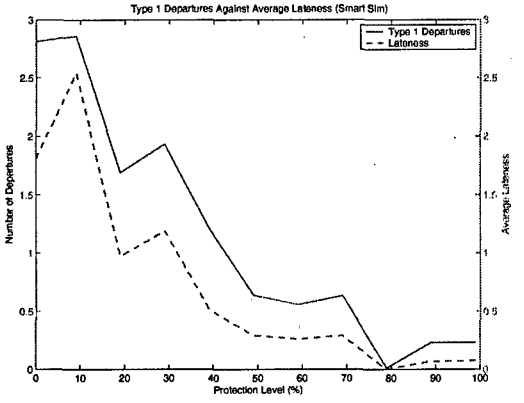
(a) Scenario 1 constraint violations



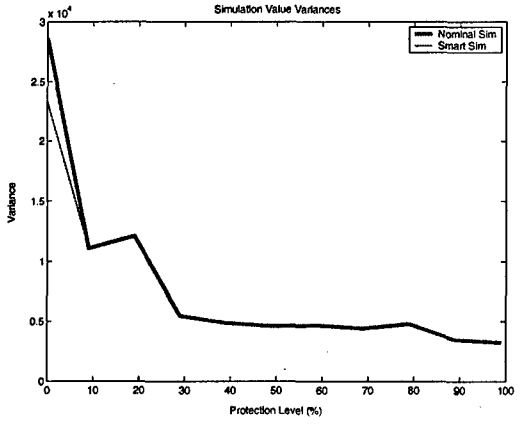
(b) Earliest departure times for Scenario 1



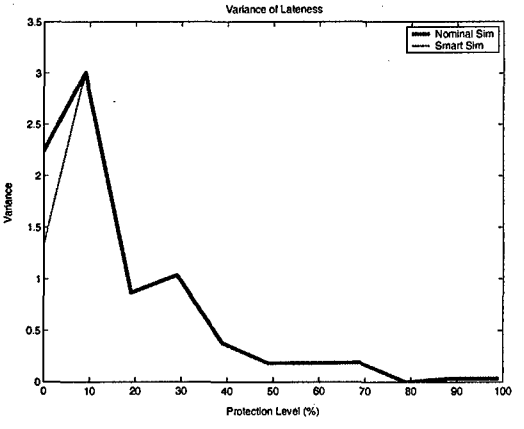
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-24: Experiment 2, Scenario 1 results

Node	Protection Levels						
	0%	10%	20%	30%	40%	80%	90%
1	-	-	-	-	-	-	-
2	I S	I		I S		S	S
3	S				I	I	I S
4	I S	I	I S		S	S	S
5	I	S	S				
6		S	I S	I S	I S	I S	I
7	I	I	I	I	I	I	I
8	I S	I S	I	I S	I S	I	I
9	-	-	-	-	-	-	-

Table 5.3: Scenario 1 Plan Itineraries

Scenario 2: We now report notable observations from the results of Scenario 2. Figure 5-25 reports a dip in the objective function value that exists in the protection range from 70% to 90%. Graph (a) of Figure 5-27 indicates fairly erratic behavior in the constraint violations. The violation lines are jagged, and Type 2 violations extend very far into the protection levels. Graph (b) reveals similarly erratic behavior with the earliest departure times. Type 1 and 3 violations spike throughout the protection levels, with both curves appearing to move in relative unison. The curve of Type 2 violations remains fairly constant at around 60 hours throughout the protection levels, while the curve representing Type 2 violations for the Smart Sim is lower in the initial protection levels but rises by protection level 20%.

When compared to those of the Base Scenario, average simulation values for Scenario 2 are substantially higher. This is expected, as the opportunity windows are relaxed in this scenario, and more tasks are accomplished on average as a result. The dip in objective function value that exists in the 70% to 90% protection range is explained by the fact that the majority of the runs did not terminate with a solution that satisfied the required optimality gap within the allowable runtime (of 5000 seconds). Figure 5-26 reveals this to be the case and reveals the large optimality gaps that still exists among the solutions. It should be noted that this suboptimality does not explain the unexpected behavior in Graphs (a) and (b) of Figure 5-27, as while the solutions might not be optimal, they are still feasible. As such, they satisfy the

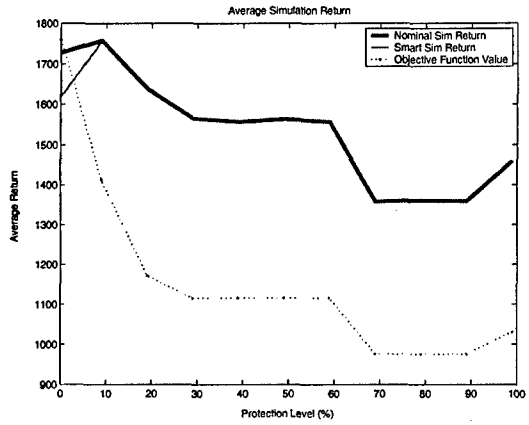


Figure 5-25: Sim and objective function values for Scenario 2

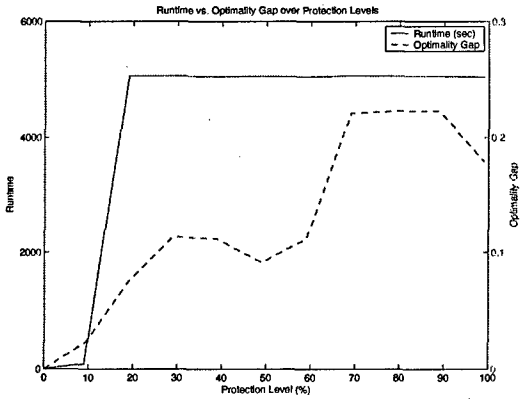


Figure 5-26: Runtime and Optimality Gap for Scenario 2

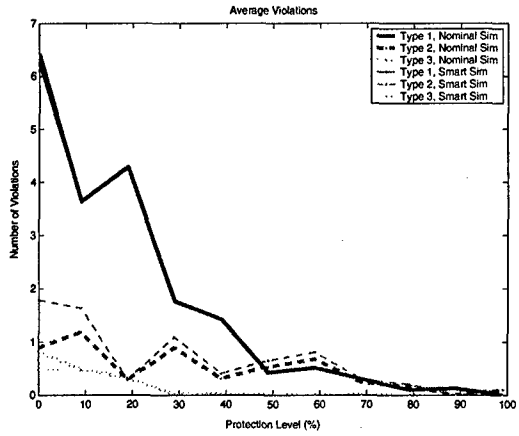
constraint requirements and the protection imposed on the constraints. As reported by Graph (b), all of these violations occur at around 60hrs into the simulation, where we placed the close of all of the opportunity windows. The fact that the UAVs are still airborne at 60 hours into the simulation (ranges for type A and B UAVs are 30 and 50, respectively) is partially due to the fact that they leave the airbase and take to the air at a time after the 0 hours start time, and partially because of the fact that our formulation does not factor loiter time into its range calculations. Loiter time, which is the flight time that is not spent travelling between targets or actively completing a task, is not considered in our formulation in order to preserve linearity in the range constraint. An interesting point is that all of the reported opportunity window violations occur not because of uncertainty in the opportunity windows and Opportunity Window constraints (there is none), but in uncertainty in the Separation constraints (uncertainty in the travel and service times). This indicates that uncertainty in some constraints can have the an impact on the feasibility of other constraints.

As reported by Graphs (a) and (b), a fair number of Type 3 violations occur in this scenario. This is due to the fact that the plans utilize the UAVs more. All Type A UAVs perform 2 strikes in the unprotected plan, and one Type A UAV performs an ISR at a target in addition to its two strikes. The Type B UAV performs ISR on all of the remaining targets in this plan. The heavy utilization of the UAVs causes tightness in the range constraints, and results in the observed Type 3 violations. It is interesting to note that this tightness is brought about by the loosening of the Opportunity Window constraints. The reason for the similar behavior of the earliest departure times of Type 1 and 3 violations is not entirely clear, but might be related to the fact that many of the same uncertain coefficients occur in both the Separation and Range Constraints (the service and travel times).

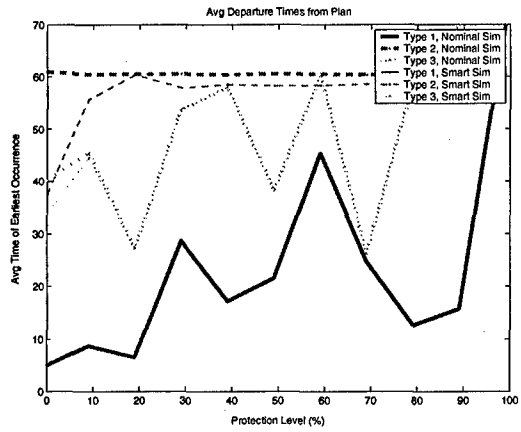
In comparison to the results for the Base Scenario, while it was unexpected that Scenario 2 solutions would experience more violations and higher variance, Type 1 and 2 violations do occur later (although it is not hard to see why this is the case for Type 2 violations, because all of the opportunity windows close later in this

scenario). It should be pointed out that the solutions to Scenario 2 call for more tasks being accomplished. This opens up the possibility for more violations, which would support the higher violation count. As pointed out earlier, the greater number of Type 3 violations highlight the interaction between constraints. Loosening opportunity windows caused more targets to be visited, made the other constraints tighter, and consequently opened the way for increased violations among these other constraints. In fact, the added tasks in the plans caused even the Opportunity Window Constraints to be tight, and increased Type 2 violations were seen as a result.

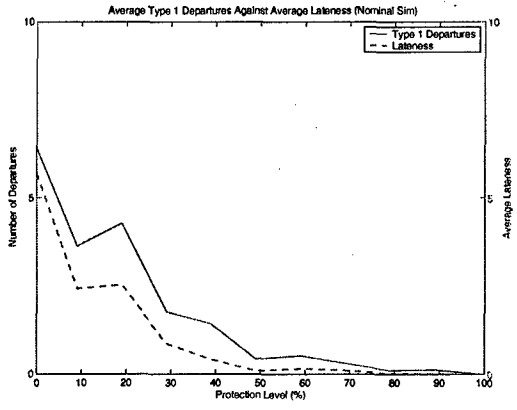
We now relate the results to the secondary hypothesis that this scenario was designed to address: that tightened windows would lead to better protected solution performance. While the results might be unexpected, they are still consistent with this hypothesis. The Base Scenario solutions did improve more with protection level than did Scenario 2 solutions. The improvement of solutions in constraint violation, earliest departure time, lateness, and metric variance as protection level was increased was much more consistent and even reported a greater magnitude. In the next scenario we see what happens as opportunity windows are smaller than in the Base Scenario.



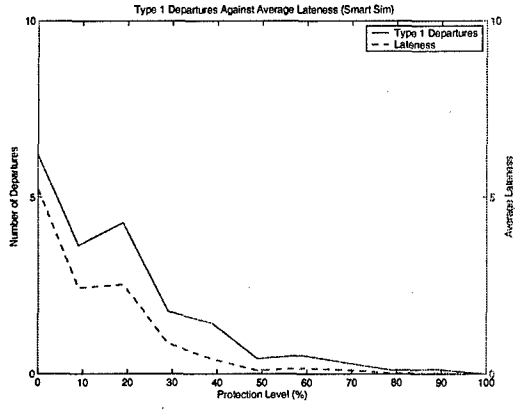
(a) Scenario 2 constraint violations



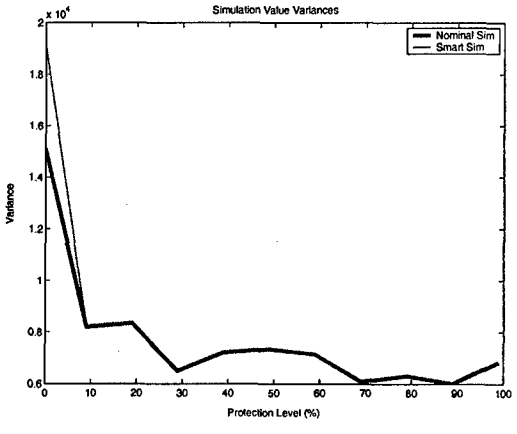
(b) Earliest departure times for Scenario 2



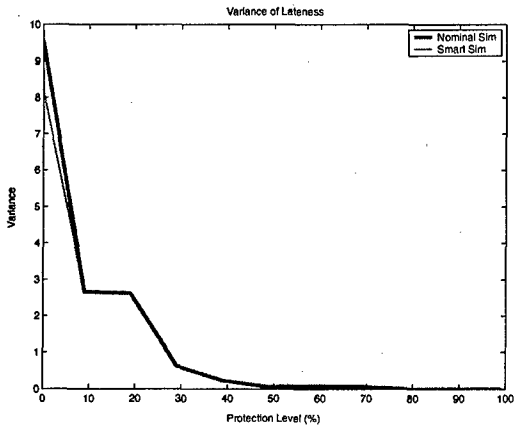
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-27: Experiment 2, Scenario 2 results

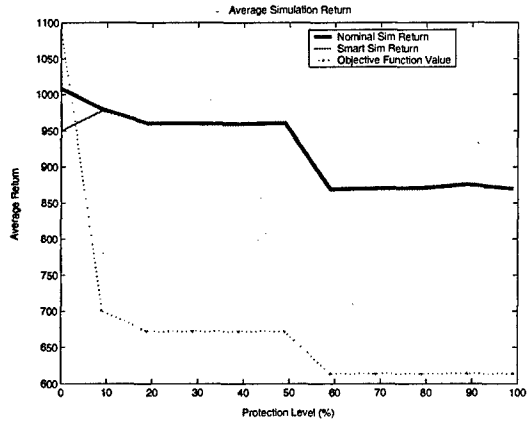
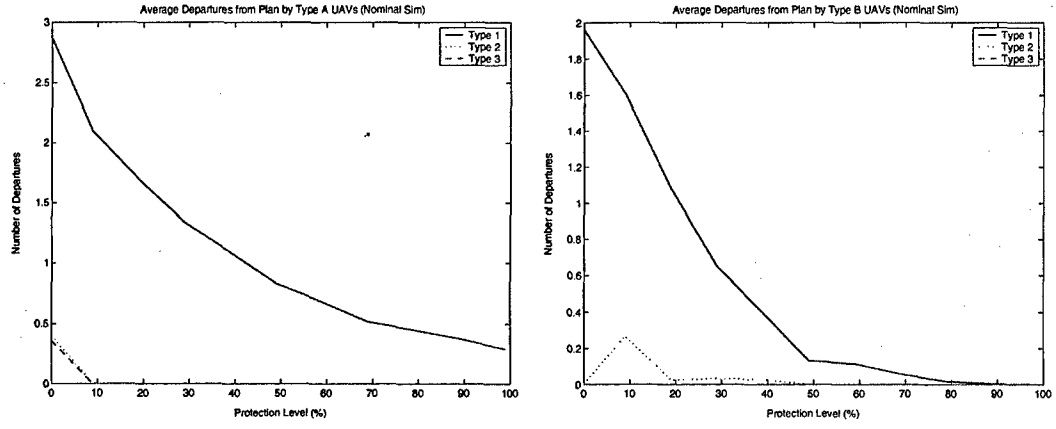


Figure 5-28: Sim and objective function values for Scenario 3



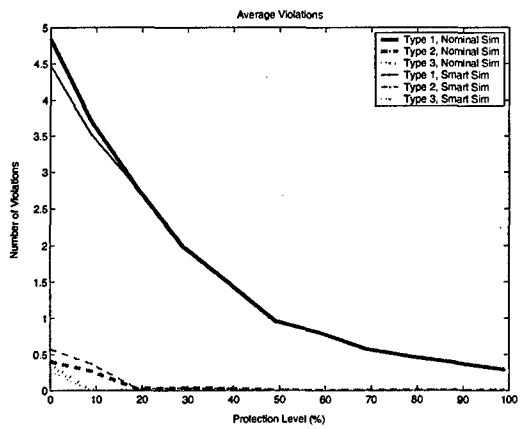
(a) Type A UAV Constraint Violation

(b) Type B UAV Constraint Violation

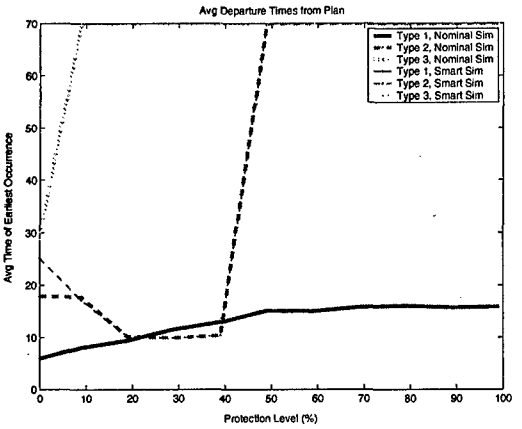
Figure 5-29: Constraint Violation by Vehicle Type

Scenario 3: Figure 5-28 reports results consistent with those previously seen. While the tighter opportunity windows of Scenario 3 cause the 0% protection objective function value to be lower than that reported in the Base Scenario, reported simulation values remain very similar in these two scenarios. Graphs of the constraint violations and earliest departure times are consistent for this scenario and report improvement in these two metrics, outside of a drop in the earliest Type 2 departure times in the 20% to 40% levels. Figure 5-29 indicates that all of these Type 2 violations are caused by the Type B UAV in the unit. Furthermore, the change in earliest departure time indicates that the particular target at which the violations

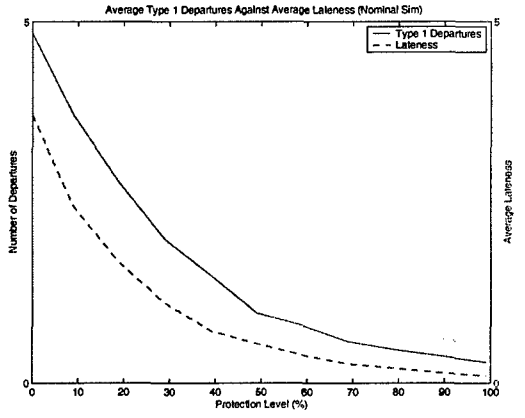
occur changes at the 20% protection level. In particular, the Type 2 violations in the 10% protection plan occur at Target 7, which the plan calls to visit at 17.53 hours. Because the close of the opportunity window for that target occurs less than 0.5 hours later, at 18 hours, the plan often violates this window. The violations for the 20% plan occur at Target 6. This plan calls for the Type B UAV to perform an ISR at the target at time 9.26, when the close of the opportunity window is at 11. While the 10% plan also calls for an ISR at the target, it plans an arrival time of 8.93 hrs, which is sufficiently far from the opportunity window to escape any violations. The delay in the arrival time at Target 6 might be due to the need to satisfy the added protection of the Separation Constraints. If this were true, it would stand as another example of the dependency between constraint types. The behavior would also suggest a conflict between protection of the different constraints. This observation also highlights the fact that the formulation does not apply protection in a manner that prevents violations that occur early in the plans more than those that occur later. While such a formulation characteristic would be favorable, it would require that protection be dependent on the constraints and arrival times.



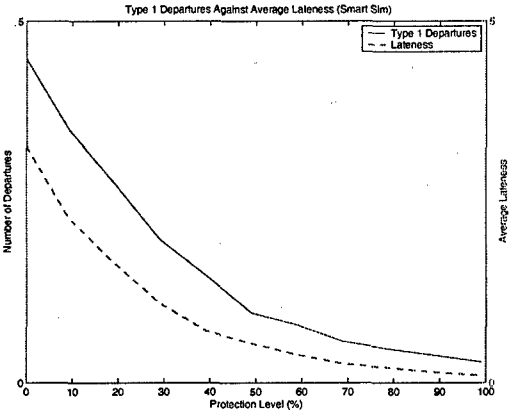
(a) Scenario 3 constraint violations



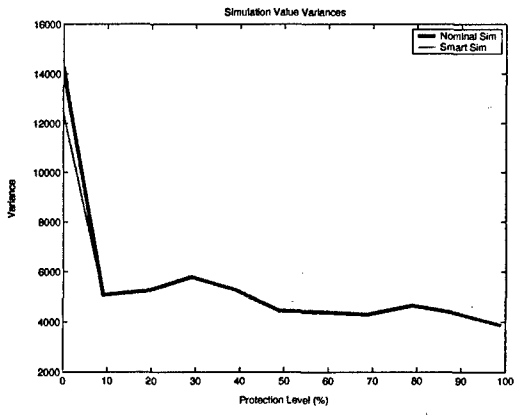
(b) Earliest departure times for Scenario 3



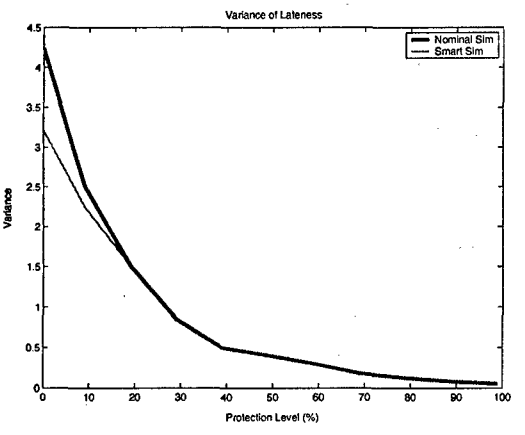
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

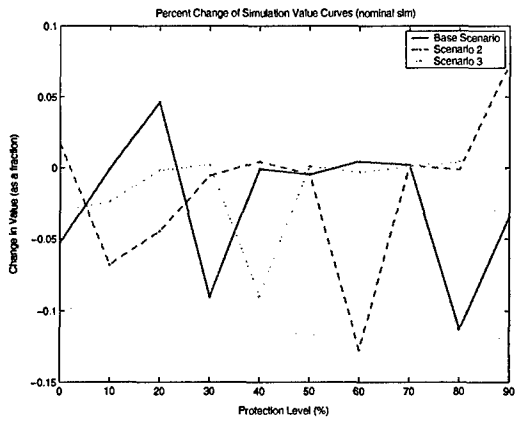
Figure 5-30: Experiment 2, Scenario 3 results

We now address the insight that the results to the foregoing three scenarios provide us with regard to the first hypothesis we test in this experiment. We feel that the results are consistent with our general belief that the protected solutions perform better as the tightness in the constraints increases. Type 2 violations are eliminated sooner in the protection levels, and results are less erratic.

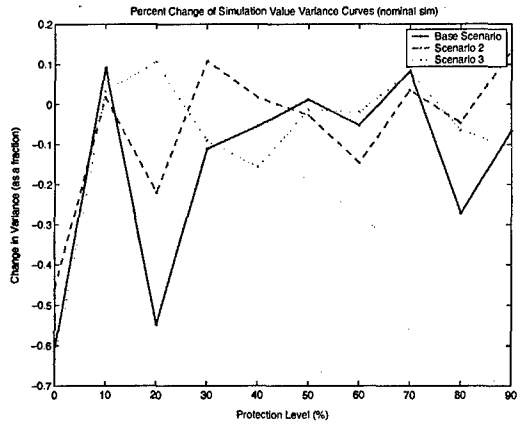
In order to further analyze the results, we calculate the percent change in each of the metrics, across the protection levels. The values are tabulated by finding the difference in the metric values in consecutive protection levels, and dividing by the value in the smaller protection level. We include plots of the results for the Nominal Sim values, Type 1 violations, lateness, value variance, and lateness variance. Better performance would be indicated by larger values in the graph of percent change in simulation value (Graph (a)), and smaller values in the remaining plots. Plots of the percent change are not reported for the other violation types, because the changes there are not significant enough. Plots for changes of the earliest departure times are omitted as well because the changes in these graphs resemble those of a step function, and results are not particularly meaningful.

The percent change plots do not appear to be conclusive. The first observation we make is that the results reported by Graphs (c) through (e) are drowned out by the large spike in the Base Scenario line at the 80% level. The percent change is so dramatic here because Type 1 violations and lateness increase from levels close to zero. Graphs (a) and (b) reveal the degree of oscillation that would be observable in the remaining graphs, had this spike not occurred. The great deal of oscillation does not invalidate our hypothesis, but the fact that the lines of the different scenarios frequently cross is problematic. Had the results been consistent with our hypothesis, we would have expected lines that did not cross paths, indicating that protected solutions to Scenario 3 were consistently outperforming the protected solutions to the other two scenarios. This is not indicated for any of the plots, and the only conclusion that can be made is that, as noted before, the results for Scenario 2 are erratic.

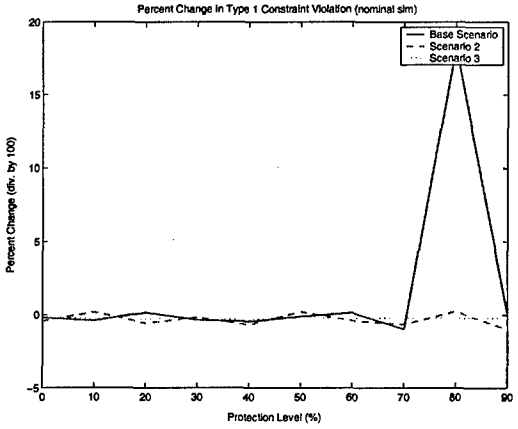
The results of the past three scenarios also make evident the fact that our belief that constraint tightness can be achieved by narrowing the target opportunity



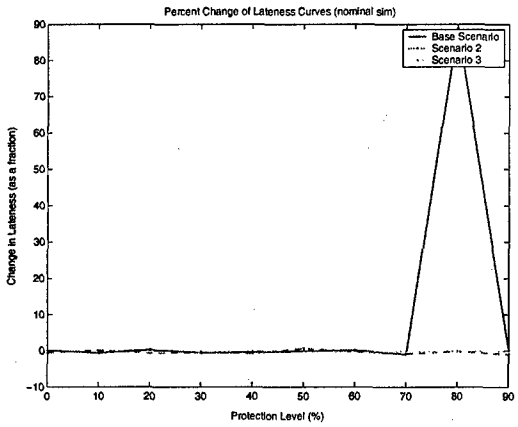
(a) Percent change in simulation value



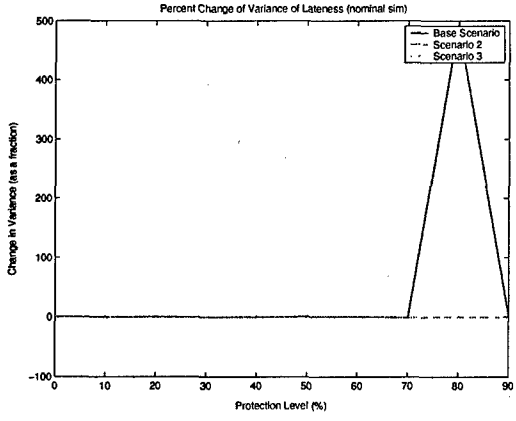
(b) Percent change in simulation value variance



(c) Percent change in Type 1 counts



(d) Percent change in lateness



(e) Percent change in variance of lateness

Figure 5-31: Percent change in metrics for Scenarios 1-3

windows was not always correct. This hypothesis overly simplifies the factors governing constraint tightness. For instance, while we predicted that the wide target opportunity windows of Scenario 2 would loosen the constraints and therefore reduce violations, the widened windows allowed the plans to utilize UAVs more aggressively. The plans instructed UAVs to visit more targets before returning to base. This caused a tightening of the separation and range constraints that led to increased violation among all of the constraint types (even Type 2 violations increased). On the other hand, while the narrow opportunity windows of Scenario 3 did force target arrival times to occur closer to the window close times, the narrow windows also acted to decrease the number of targets that could feasibly be visited. Fewer opportunity windows could be violated, the fewer targets visited brought more slack to the separation and range constraints, and, in general, fewer violations occurred.

The foregoing experiments also revealed that increasing solution protection does not always ensure improved metric performance (our Base Hypotheses). This was obviously the case in simulation value, which almost always decreased with protection, but was true even among the other metrics. As observed in Scenario 3, it is possible for solutions with added protection to experience violations that occur sooner in the plans. Furthermore, improvements in the number of constraint violations and variances were not always monotonic, and increased protection occasionally coincided with worsened metric values. It was observed, in general, that this worsening of metric values occurred among the lower protection levels; metric improvement stabilized in the higher protection levels.

Scenario 4: Scenarios 4 and 5 begin our analysis of the second hypothesis we address in this experiment. In Scenario 4, the separation between the target opportunity windows is increased over the Base Scenario. When compared to the results of the Base Scenario, the objective function value is higher across all protection levels, but experiences the same initial drop at the 10% protection level. The simulation values in Scenario 4 are affected more by this drop than are those in the Base Scenario, as they lose 25% of their value within this range.

In terms of constraint violations, both the Base Scenario and Scenario 4 have

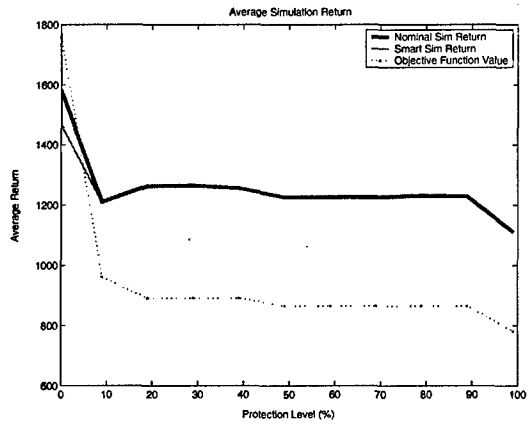
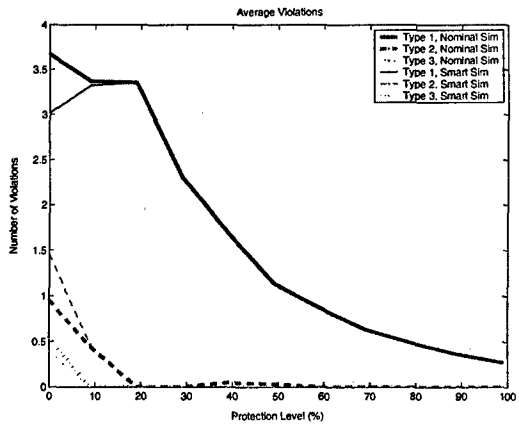


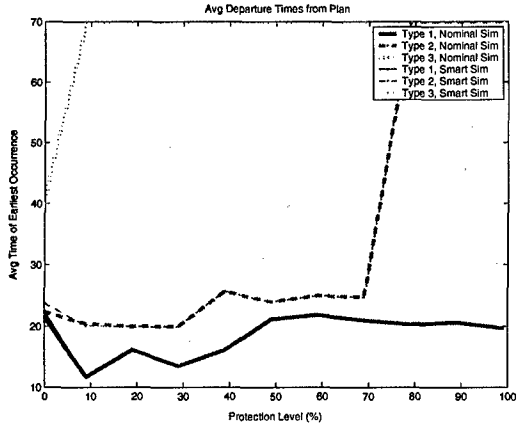
Figure 5-32: Sim and objective function values for Scenario 4

nearly identical Type 3 violation curves. The Type 2 curves resemble one another as well, but Scenario 4 reports more Type 2 violations at the 10% level and less at the higher protection levels. Scenario 4, however, reports substantially fewer Type 1 violations in the initial protection levels (greater than 25% fewer at the 0% protection level). This difference is even more significant when considering the fact that Scenario 4 plans visit more targets and have more potential for Type 1 violations.

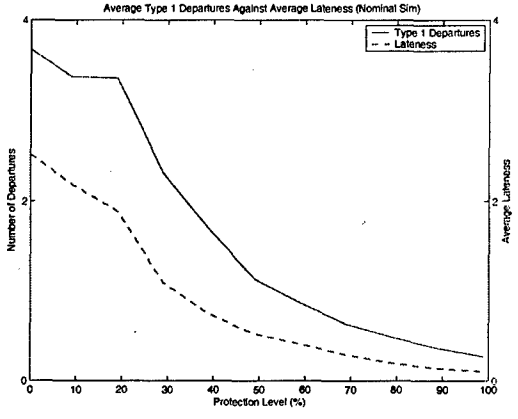
The graph of Scenario 4 departure times indicates later departure times for all constraint types. In terms of Type 2 departure times, this is explainable by the fact that, in the process of increasing separation, Scenario 4 had later opportunity window close times, on average. While the curves for the Scenario 4 departure times of Type 2 and 3 violations are smooth and resemble those of the Base Scenario, the curve for Type 1 violations experiences a fair amount of oscillation in the early protection levels.



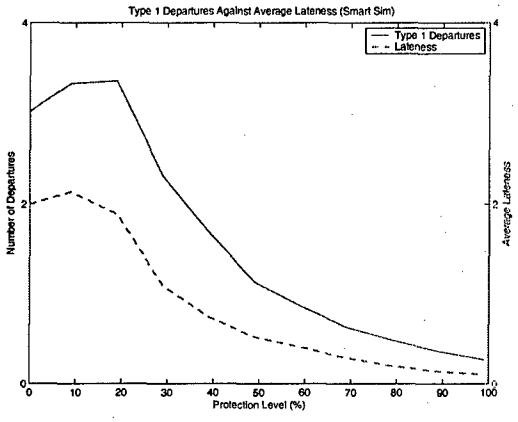
(a) Scenario 4 constraint violations



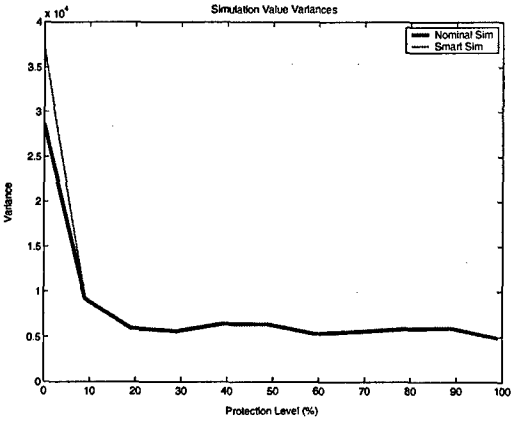
(b) Earliest departure times for Scenario 4



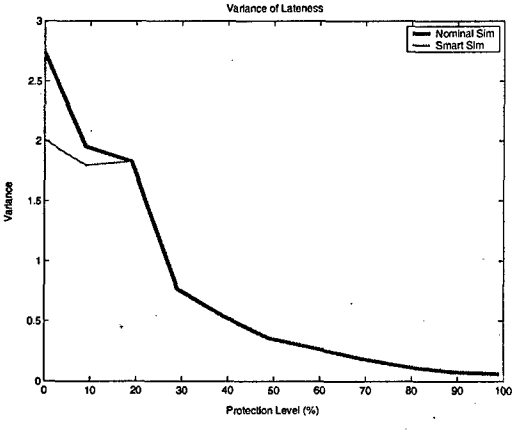
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-33: Experiment 2, Scenario 4 results

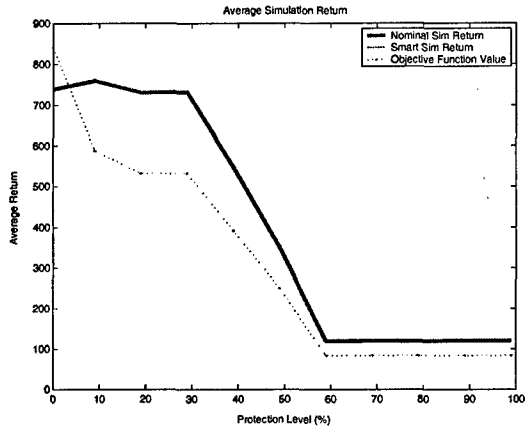


Figure 5-34: Sim and objective function values for Scenario 5

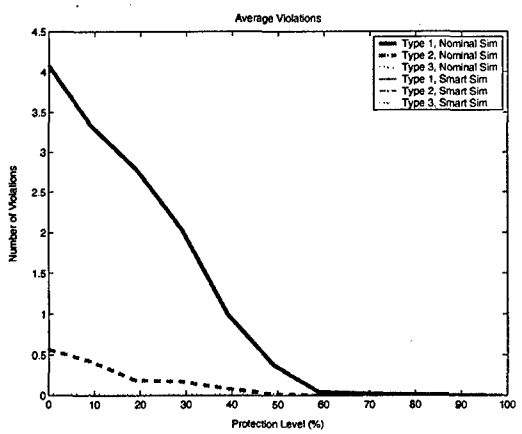
Scenario 5: In this scenario, Base Scenario target opportunity windows are altered so that a complete overlap exists between the windows. As expected, Figure 5-34 reports a substantial decrease in the objective function and simulation values from those reported in the Base Scenario. The initial drop in objective function value over the protection levels is much more gradual than in the Base Scenario, but a steep drop is experienced in the 30% to 60% protection level range, and objective function value bottoms at approximately 100 by the end of this range. While simulation value experiences the drop in this protection level range, it actually increases at the 10% level.

Graph (a) of Figure 5-35 reports that no Type 3 violations occur in any of the protection levels. This result is intuitive, because, as seen in Figure 5-34, the overlapping opportunity windows prevent the plans from visiting many targets. As a result, range constraints are not binding, and no Type 3 violations occur. Even though few targets are visited, the average Type 1 violation count is high relative to the other target opportunity window separation configurations. While this average is high in the unprotected solution, it drops rapidly, and Type 1 violations disappear completely in the 100% protection level. This behavior has not been observed until now in the scenarios!

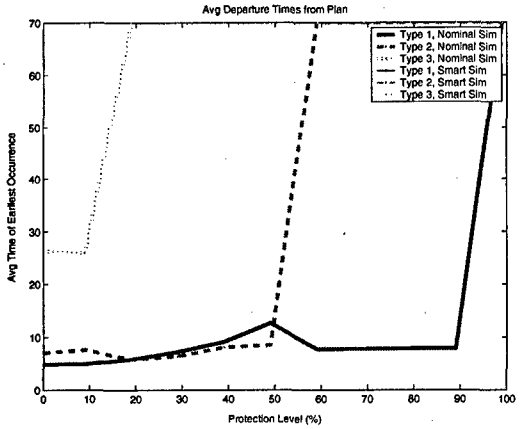
Graph (b) reports that departure times occur earlier for Scenario 5 in both Type 2 and 3 violations. While the curve for Type 2 departure times remains fairly flat

in the first few protection levels, it jumps at the 60% protection level, where Type 2 violations disappear. This occurs sooner than in both Scenario 4 and the Base Scenario (where Type 2 violations disappear at the 80% protection level). Graphs (c) through (f) of Figure 5-35 report curves for lateness and the variances that have relatively steep drops with respect to protection level. Values for these metrics are very low at the 100% protection level (and even reach 0), and these resting points are reached earlier in the protection levels.

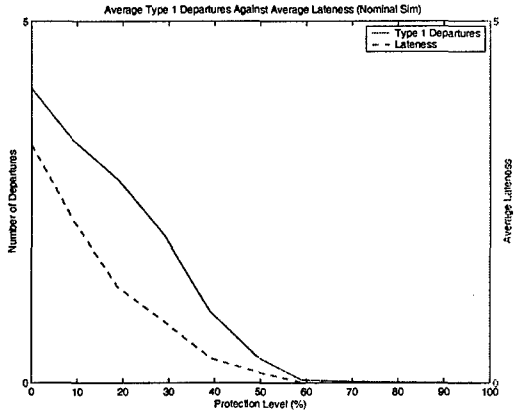
Results for Scenarios 4 and 5 confirm the hypothesis that protection becomes more beneficial as target opportunity windows overlap. Against our expectations, however, instead of an increase in Type 2 violations, the increase in overlap mainly affected Type 1 violations. At any rate, not only did the added violations cause solutions to improve as protection increased, but the rate of this improvement was steeper in many cases. While the lines representing the percent change in the metrics for the different scenarios do cross occasionally in Figure 5-36, that the percent change in Scenario 5 consistently outperforms that of the others is evident. Scenario 5 clearly demonstrates an instance where incorporating protection into the formulation is beneficial. Not only did this scenario experience higher improvements with protection in constraint violation, departure times, lateness, and variance, but it even experienced an improvement in simulation value when protection was initially applied. Scenario 5 marks the first occasion that improvements in all metrics were observed at a protection level.



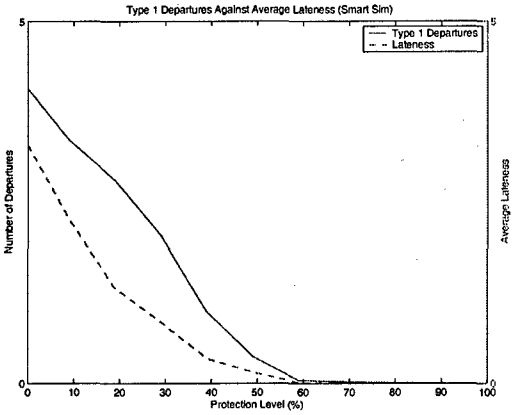
(a) Scenario 5 constraint violations



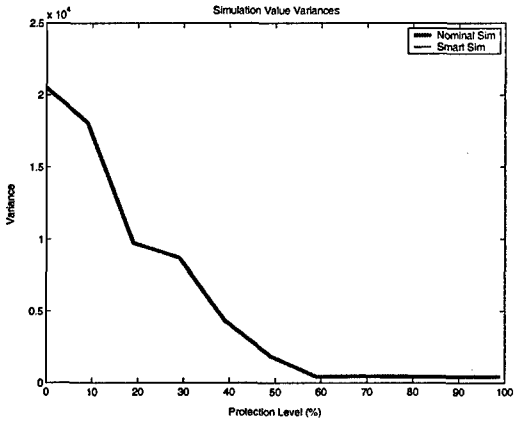
(b) Earliest departure times for Scenario 5



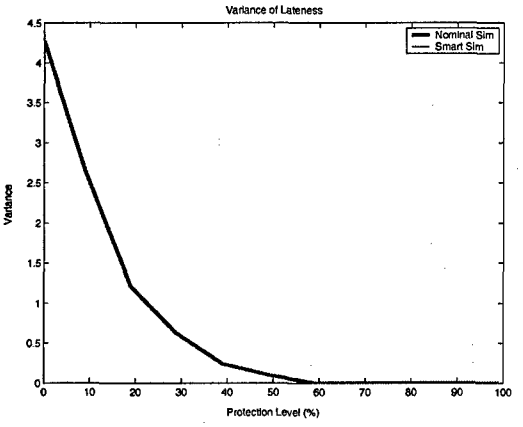
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim

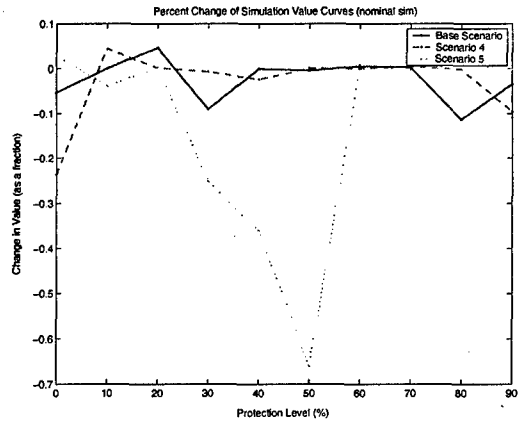


(e) Variances of sim value

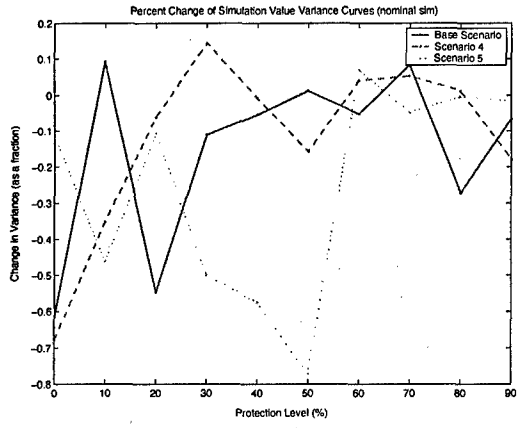


(f) Variance of the lateness statistic

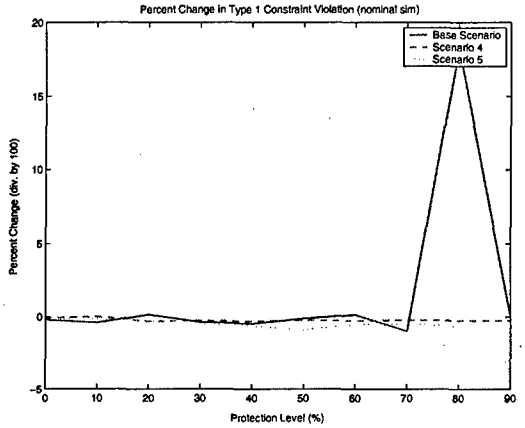
Figure 5-35: Experiment 2, Scenario 5 results



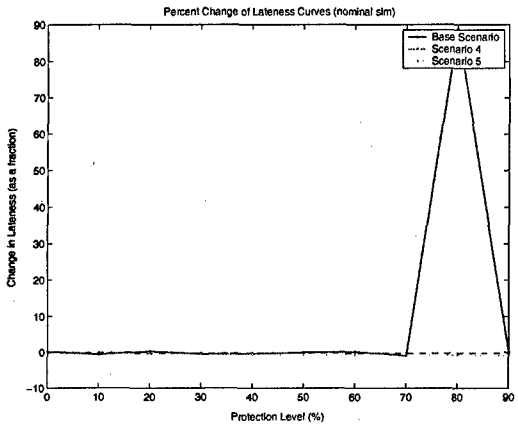
(a) Percent change in simulation value



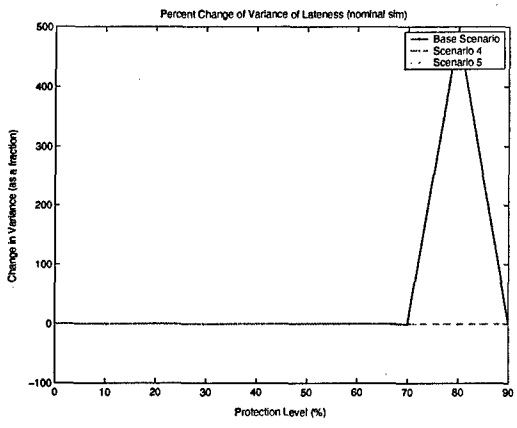
(b) Percent change in simulation value variance



(c) Percent change in Type 1 counts



(d) Percent change in lateness



(e) Percent change in variance of lateness

Figure 5-36: Percent change in metrics for Scenarios 4 and 5

5.4.4 Experiment 3

Motivation

In this experiment, we test the performance of protection as the service times are perturbed. Our belief is that increasing service times will act to tighten the Separation Constraints and cause more violations of the unprotected solutions, thus making protection more beneficial. We test two scenarios, one with no service times and one with large service times, and compare results to these scenarios with the Base Scenario tested in Experiment 2.

Hypotheses

- On average, protected solutions perform better as service times increase.

Scenarios Tested

1. No task service times; and
2. Large task service times.

Results and Analysis

Scenario 1: This scenario tests performance of protected solutions in an environment where no service times exist. The expectation was that this change would decrease the number of Type 1 violations that the plans encounter. Surprisingly, Type 1 violations actually increase over the Base Scenario, as do Type 2 violations. Type 3 violations are almost nonexistent throughout the protection levels. The lateness values are also high, and the average Type 1 violation occurs by close to 2 hours. Figure 5-37 reports higher simulation values across all protection levels, and also indicates a spike in the objective function value characteristic of the solver's inability to find the optimal solution within the time limit. Figure 5-38 confirms that high optimality gaps occurred throughout the protection levels.

Analyzing the results, it is evident that the elimination of service times allows plans to schedule more tasks (indicated by the higher objective function values). In

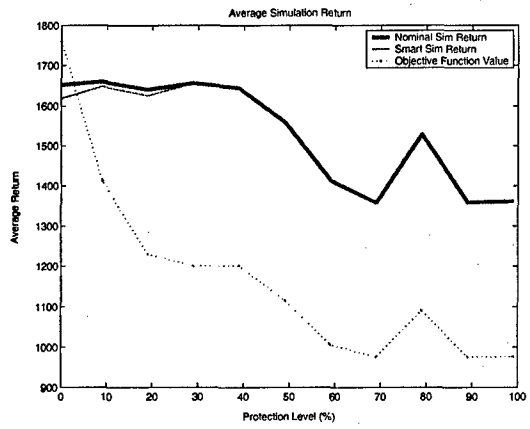


Figure 5-37: Sim and objective function values for Scenario 1

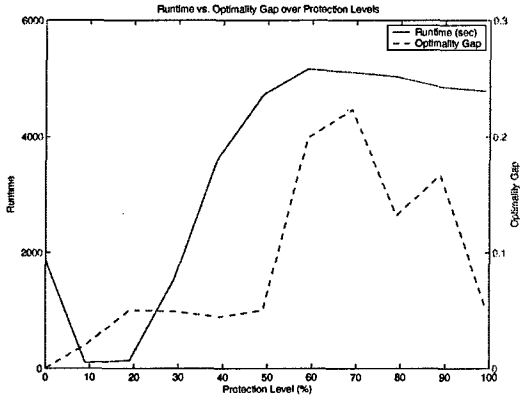
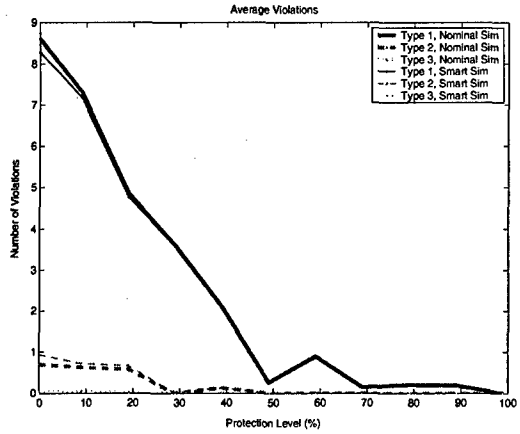
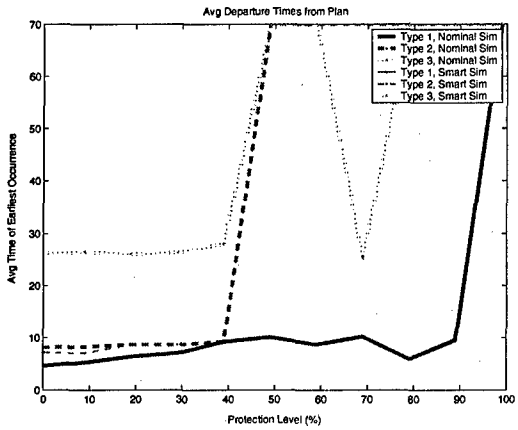


Figure 5-38: Runtime and optimality gap for Scenario 1

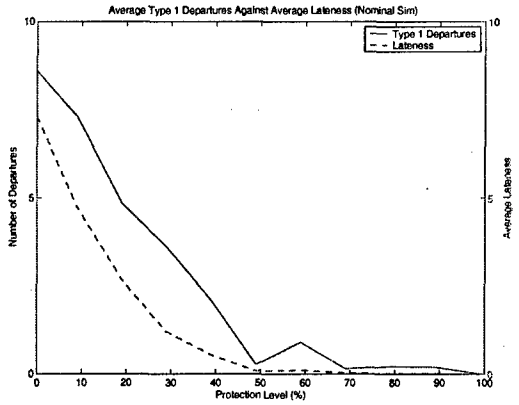
order to satisfy the target Opportunity Window Constraints, arrival times are scheduled closer to one another, causing more Type 1 violations. Opportunity Window Constraints are also tighter as a consequence, a fact reflected by the Type 2 violations. Few Type 3 violations occur because the nonexistent service times allow plans to schedule the tasks well before range limitations are reached.



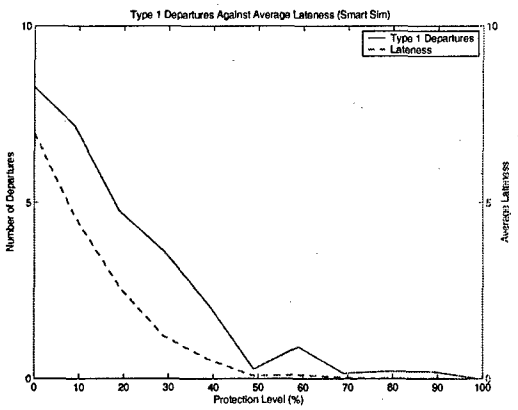
(a) Scenario 1 constraint violations



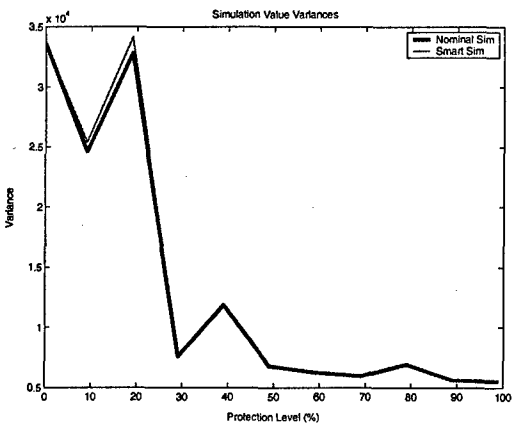
(b) Earliest departure times for Scenario 1



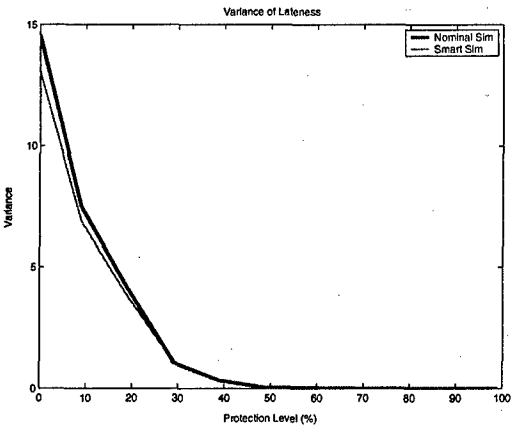
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-39: Experiment 3, Scenario 1 results

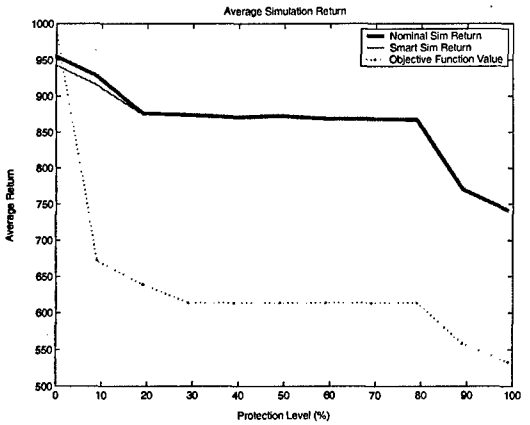
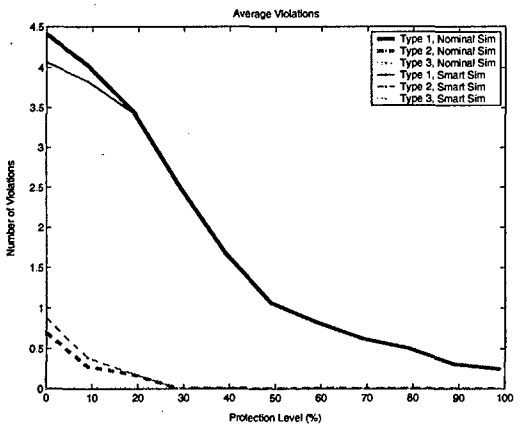


Figure 5-40: Sim and objective function values for Scenario 2

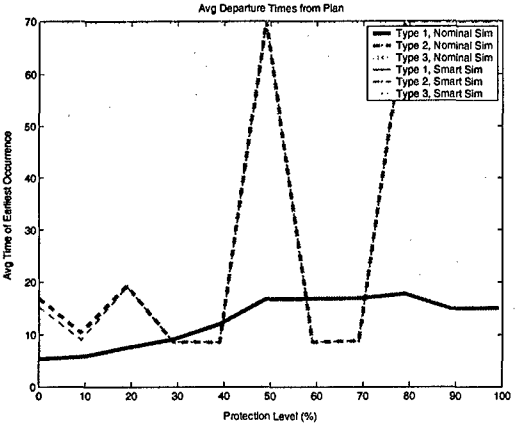
Scenario 2: This scenario tests results for a situation in which service times are increased over their levels in the Base Scenario. As expected, Figure 5-40 reports that, because of these larger service times, plans accomplish much less, on average. An effect of this is that fewer Type 2 violations occur, on average, as fewer are possible.

Graph (b) indicates the nonmonotonic departure time behavior observed earlier, and the remaining graphs again indicate smaller values, on average, for the remaining metrics.

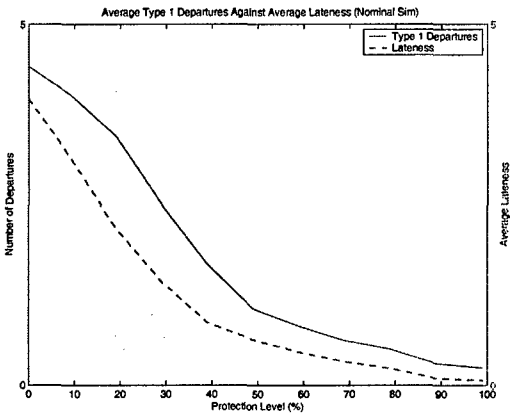
Results to this experiments' scenarios do not corroborate with our hypothesis that added protection is more beneficial as service times increase. Plots of the percent change in the metrics are similarly inconclusive with regard to this hypothesis, and report the erratic behavior of the scenario.



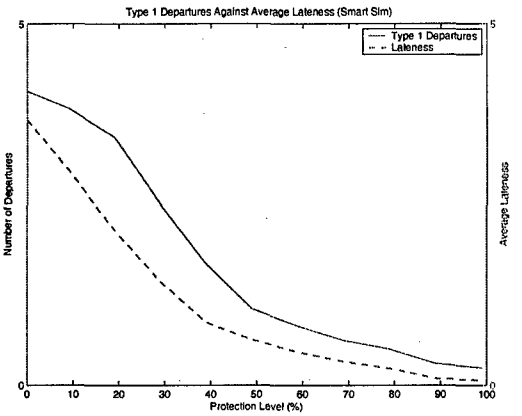
(a) Scenario 2 constraint violations



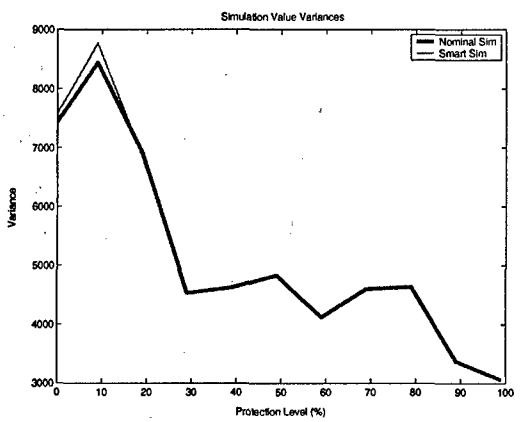
(b) Earliest departure times for Scenario 2



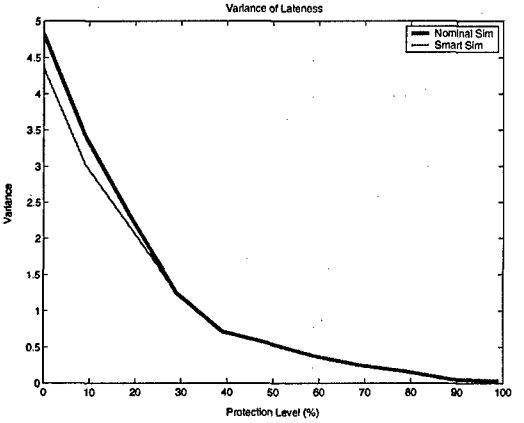
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim

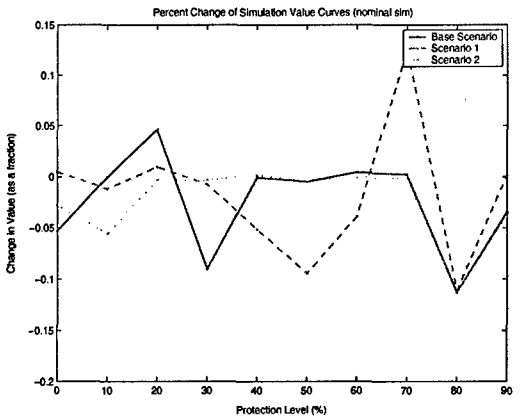


(e) Variances of sim value

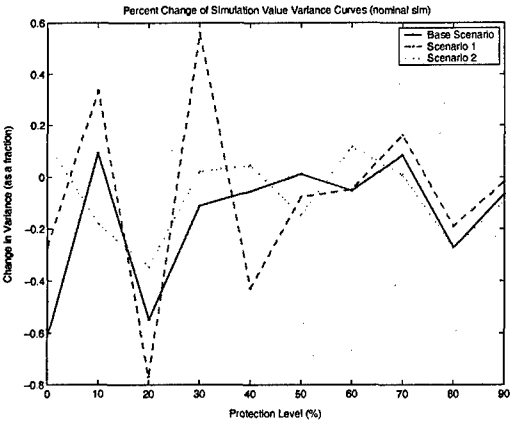


(f) Variance of the lateness statistic

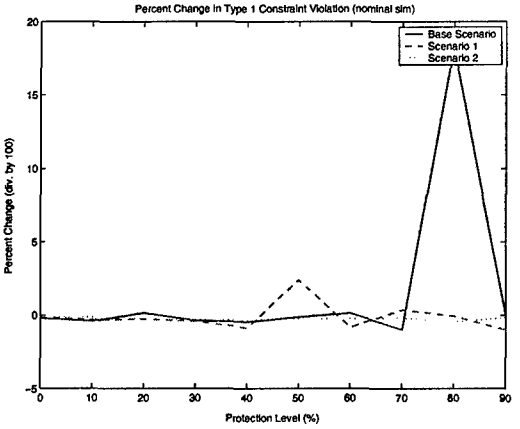
Figure 5-41: Experiment 3, Scenario 2 results



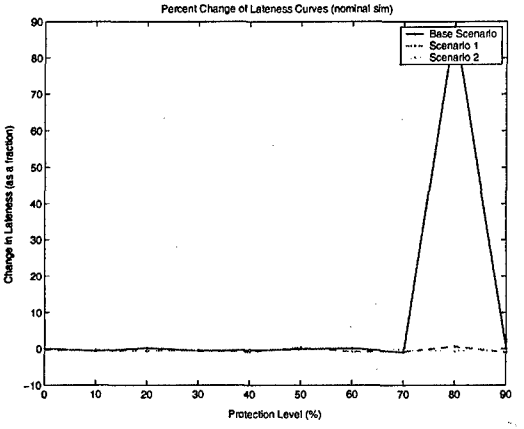
(a) Percent change in simulation value



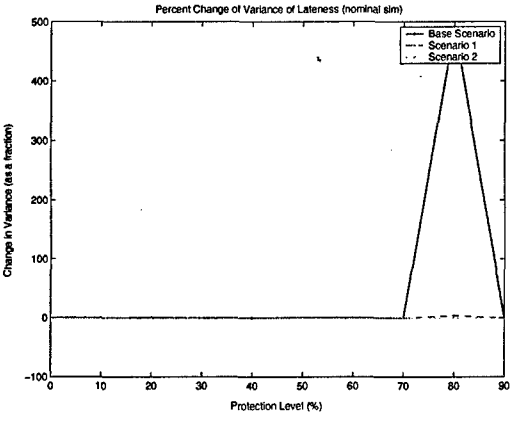
(b) Percent change in simulation value variance



(c) Percent change in Type 1 counts



(d) Percent change in lateness



(e) Percent change in variance of lateness

Figure 5-42: Percent change in metrics for Scenarios 1 and 2

5.4.5 Experiment 4

Motivation

In this experiment, we test the performance of protection as travel times are perturbed. Our belief is that increasing travel times will act to tighten the separation constraints and cause more violations of the unprotected solutions, thus making protection more beneficial. We test two scenarios: one with small travel times (half of Base Scenario values) and one with large travel times (1.5 times Base Scenario values), and compare the results of these scenarios to those of the Base Scenario tested in Experiment 2.

Hypotheses

- On average, protected solutions perform better as inter-target travel times increase.

Scenarios Tested

1. Small travel times; and
2. Large travel times.

Results and Analysis

Scenario 1: This scenario tests the performance of solutions when the inter-target travel times are reduced from their levels in the Base Scenario. To prepare the data file for Scenario 1, all travel times in the Base Scenario are multiplied by 0.5, and the remaining values are left unaltered. As reported by Figure 5-44, this change in travel times causes reported objective function values to increase over their levels in the Base Scenario, and allows simulations to achieve higher values, on average. Unexpectedly, increasing protection actually improves average simulation value. Outside of an initial dip by the Nominal Sim at the 10% level and a slight dip for both simulations at the 30% level, both simulations report increases in average value up to the 40% protection level.

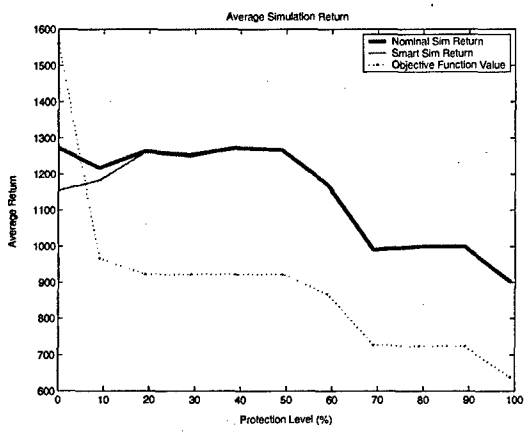


Figure 5-43: Sim and objective function values for Scenario 1

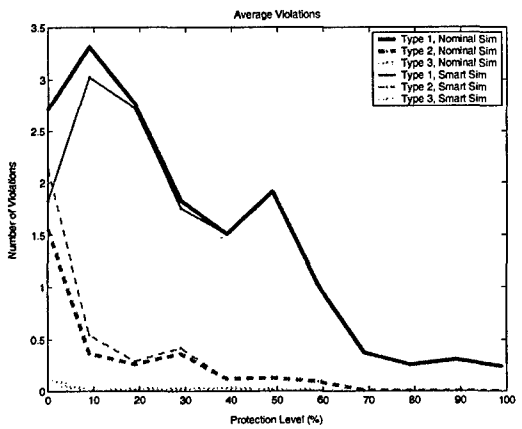
Graph (a) of Figure 5-44 indicates that the average number of Type 2 violations is closer to the average number of Type 1 violations than was usually the case in previous scenarios. All violations decrease on average, although Graph (a) indicates two increases in Type 1 violations at the 10% and 50% protection levels. These increases are manifested in average lateness, as well as the variance of average lateness (Graphs (c), (d), and (f)).

The unexpected improvement in simulation value with added protection appears to be due to the same reasons that caused the improvement in Scenario 5 of Experiment 2. Namely, the decrease in travel time allows plans to schedule more tasks. This actually increases tightness among Separation and Opportunity Window Constraints, and prompts more violations among these constraints. The Type 2 violations prevent unprotected plans from realizing rewards for many of the tasks they schedule, and while protected plans schedule fewer tasks, they are able to accomplish more of them.

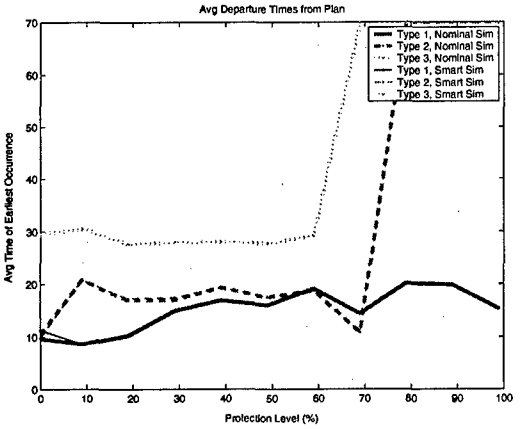
We address the two increases in the average number of Type 1 violations at the 10% and 50% protection levels. The first may be considered as a tradeoff between Type 1 and Type 2 violations, as the increase in Type 1 violations is accompanied by a decrease in Type 2 violations. Arrival times that were once outside of the target opportunity windows now appear within the opportunity window but still occur after the scheduled arrival times. The increase in Type 1 violations at the 50% level is not accompanied by a decrease in Type 2 violations. While most arrival times occur

earlier in order to safeguard against Type 2 violations, the arrival time at Target 7 is delayed. While this is a MAU target and therefore has less uncertainty associated with it, the arrival time occurs closer to the close of the opportunity window at that target, and Type 2 violations are still raised in some of the simulation runs.

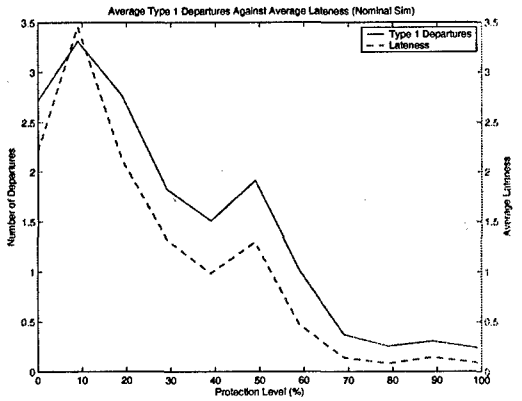
While the number of Type 3 violations is substantially lower initially than it was for the Base Scenario, as noted earlier, Type 3 violations persist later into the protection levels. The decreased travel times mean that UAVs consume less of their available range in each target that they visit, but also allow the plans to schedule each UAV to visit more targets. Until protection level 60%, plans called for 2 of the 3 Type A UAVs to visit 2 targets apiece, whereas no type A UAVs visit more than 1 target once protection is added to the Base Scenario solutions. Because this added utilization persists until the 60% protection level in Scenario 1, Type 3 violations continue until this protection level as well.



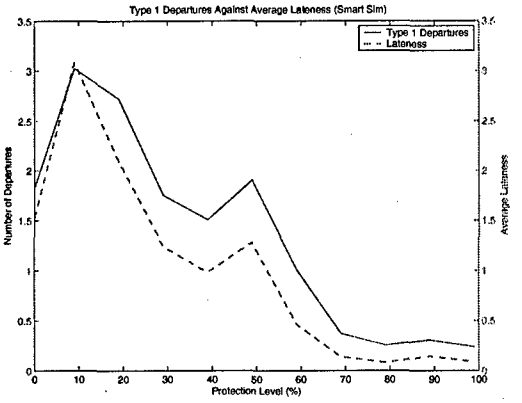
(a) Scenario 1 constraint violations



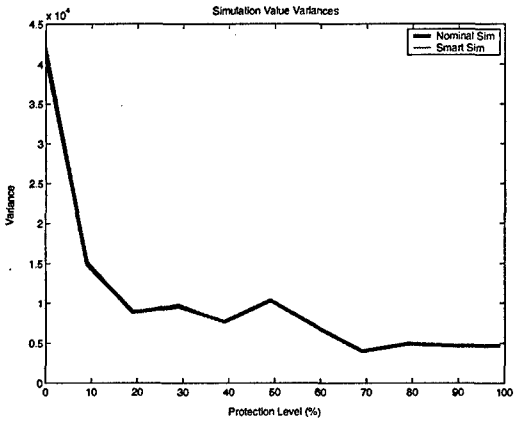
(b) Earliest departure times for Scenario 1



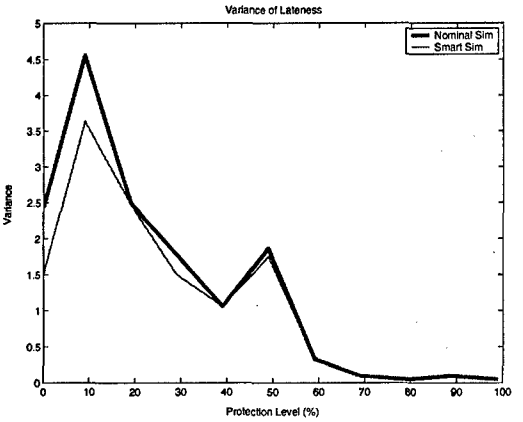
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-44: Experiment 4, Scenario 1 results

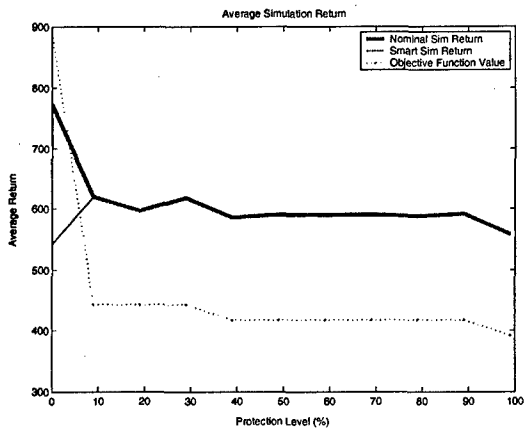
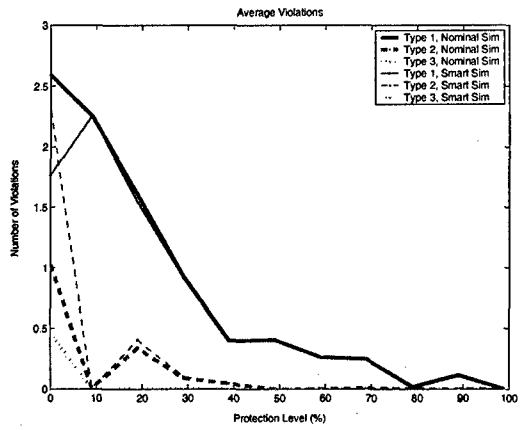


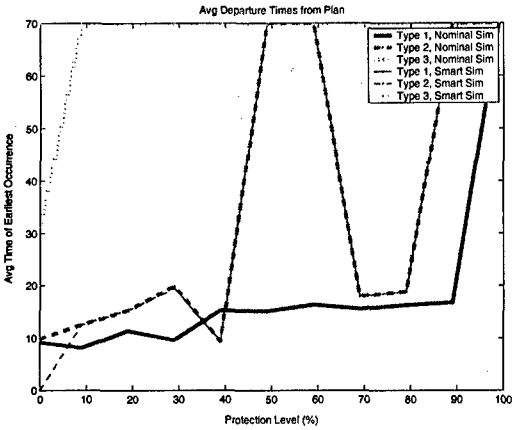
Figure 5-45: Sim and objective function values for Scenario 2

Scenario 2: In this scenario, Base Scenario travel times are multiplied by 1.5. As expected, this prevents plans from performing all of the tasks that were feasible in the Base Scenario. Unlike Scenario 1, however, average Nominal Sim value does not increase with protection level, but rather decreases. While Type 1 violations for Scenario 2 do not increase as they do in Scenario 1 (Graph (a) of Figure 5-46), Type 2 violations do increase at the 20% protection level. All Type 2 violations occur as a result of the Type B UAV missing target windows, and the plan at the 20% protection level calls for later arrivals for almost all of the targets that the UAV visits, ostensibly because of the added Separation Constraint protection. The proximity of these arrival times to the opportunity window close times and the propensity of this plan for Type 2 violations prompts the increase in simulation value variance reported by Graph (e).

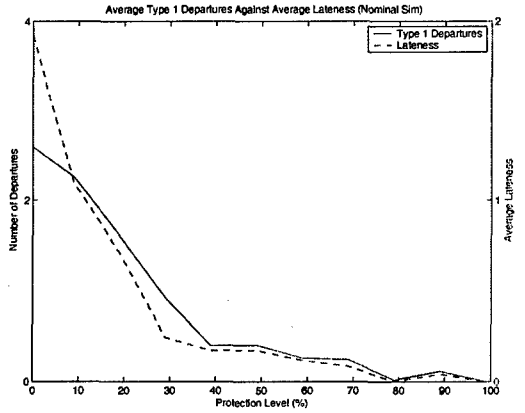
The two scenarios tested in this experiment once again challenge our beliefs concerning what dictates constraint tightness. Our belief was that tightness would increase in at least the Separation Constraints as travel times increased. Though travel times increased two-fold between Scenario 1 and the Base Scenario, little to no differences could be identified in their results. In addition, Type 1 violations decreased in Scenario 2, when they were expected to increase because of the larger inter-target travel times. This is partially due to the fact that the Scenario 2 plan performs fewer tasks and therefore has fewer opportunities for Type 1 violations, and the ratio of violations to tasks performed is similar for both scenarios. Scenario 2 experiences Type



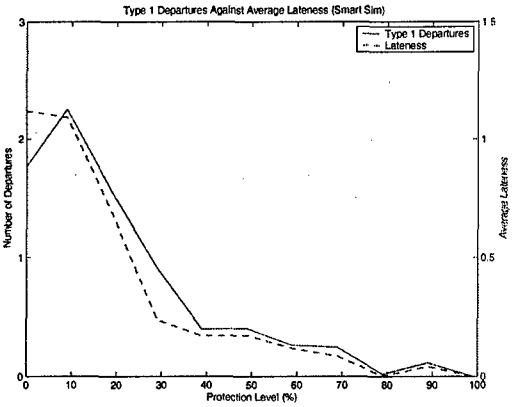
(a) Scenario 2 constraint violations



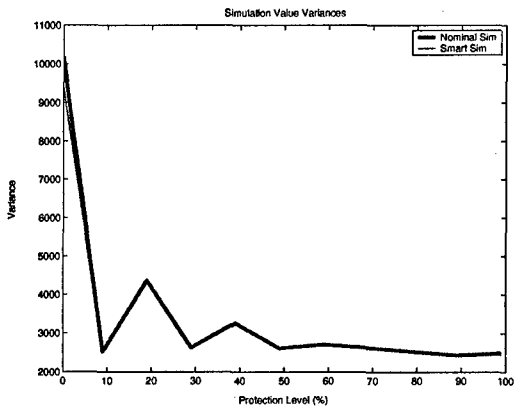
(b) Earliest departure times for Scenario 2



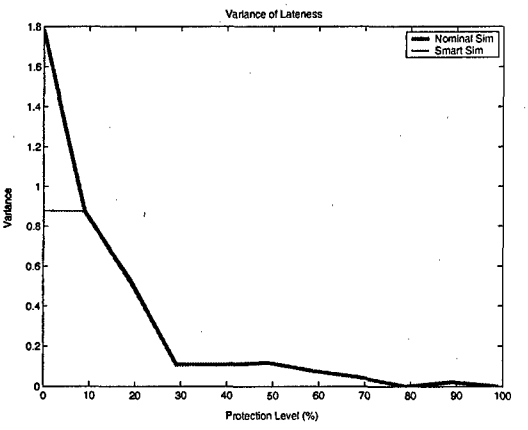
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

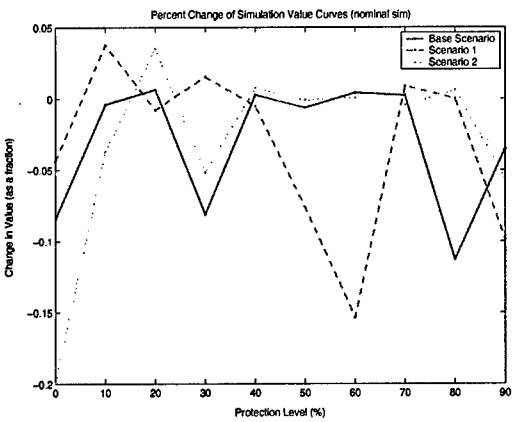
Figure 5-46: Experiment 4, Scenario 2 results

1 violations in $\sim$ 36% of the tasks it performs ($\sim$ 2.5 violations for the 7 scheduled tasks), and the Base Scenario experiences $\sim$ 38% on average (3.5 of the 9 scheduled tasks). While this is the case, our underlying belief still does not hold, as we expected there to be more violations in Scenario 2.

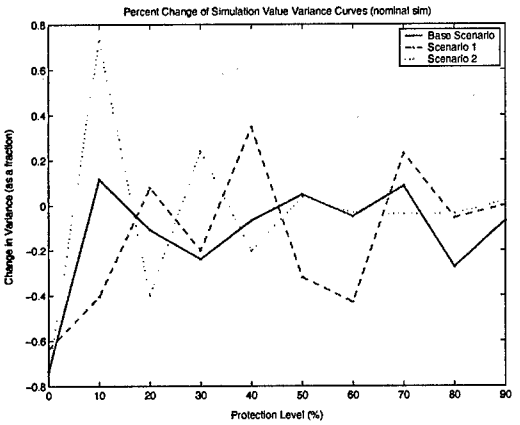
We again point out the dependence of the constraints on one another. While reducing travel times may have relaxed the separation constraints, they prompted the plans to schedule more tasks, tightening the opportunity window constraints and causing more violations among them. While constraint tightness then seems to have played a role in the Type 2 violations, it is also possible that the violations were a result of the uncertainty in the scenario. Scenario 1 travel time hat values were not altered from their state in the Base Scenario, but their size relative to the nominal travel times increased because the nominal travel times decreased. We investigate the influence of hat values and uncertainty on performance in the next experiment.

Because no clear trend in tightness existed between the scenarios, neither can any trend in the performance of protection in the solutions be established. None of the graphs in Figure 5-47 indicate clear differences in the performance of protection among the scenarios. In Graphs (c) through (e), the spike at the 80% level in the line for Scenario 2 is caused by the increase in the average number of Type 1 violations at this level. Because the number of violations at the 70% level was very small, the percent change appeared to be very large.

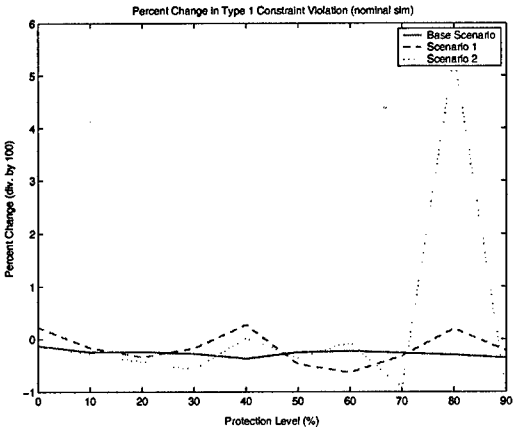
The two scenarios in this experiment and the Base Scenario of Experiment 2 indicate that a relatively small addition of protection into the solution eliminates many occurrences of Type 2 and 3 violations. The percent change in Type 2 violations increases as the number of these violations increases in the unprotected solution. In this sense, scenarios that are more prone to violation are benefitted more from protection, and our underlying belief holds.



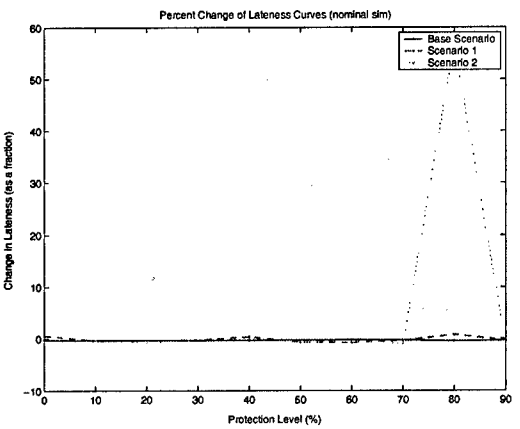
(a) Percent change in simulation value



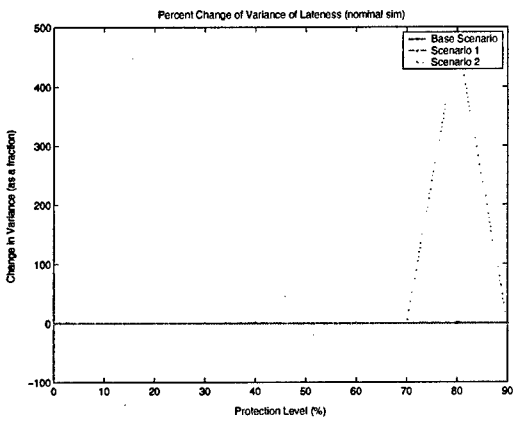
(b) Percent change in simulation value variance



(c) Percent change in Type 1 counts



(d) Percent change in lateness



(e) Percent change in variance of lateness

Figure 5-47: Percent change in metrics for Scenarios 1 and 2

5.4.6 Experiment 5

Motivation

We now test the performance of protection as the hat values are altered. According to our expectations underlined at the beginning of this chapter, increasing the uncertainty (increasing the hat values) should result in an increase in constraint violations for the unprotected solutions, and should cause protection to be more beneficial. For our initial testing, we included two scenarios: one where hat values were decreased from their base levels, and one where hat values were increased.

Hypotheses

- On average, protected solutions perform better as hat values increase.

Scenarios Tested

1. Small hat values; and

2. Large large hat values.

Results and Analysis

Scenario 1: In this scenario, hat values are decreased substantially from their levels in the Base Scenario. The objective function, service time, and opportunity window hat values associated with MAUs assume a value equal to 8% of the base value, range hat values are given 10% uncertainty, and all other hat values take on 15% of the base values. Before reporting the results to this scenario, we refer to Figure 5-48, which is a graph of constraint violation for a scenario where no uncertainty is present. As expected, solutions to this scenario are devoid of violations throughout the protection range, outside of Type 2 violations in the Smart Sim. These violations are raised when the Smart Sim chooses to skip a target whose slated arrival time occurs near the close of its opportunity window, and not because of late target arrivals on the part of the UAVs in the simulation.

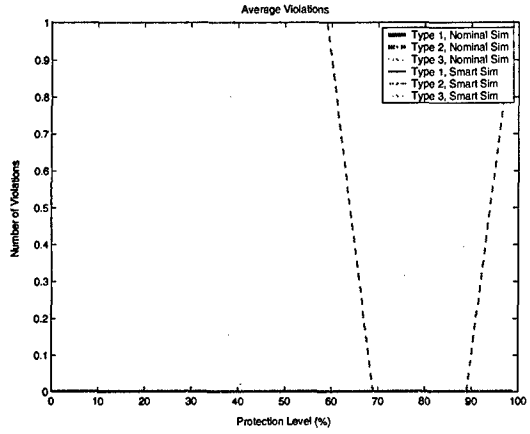


Figure 5-48: Constraint violation when no uncertainty exists

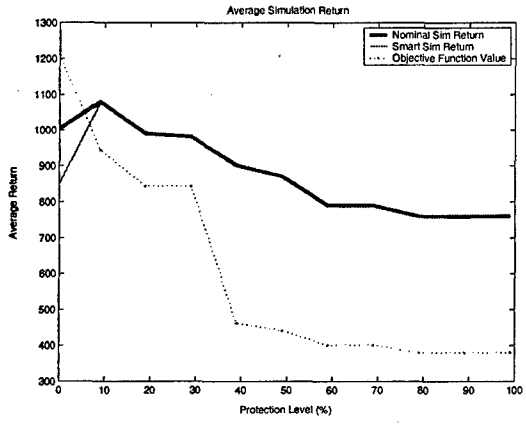
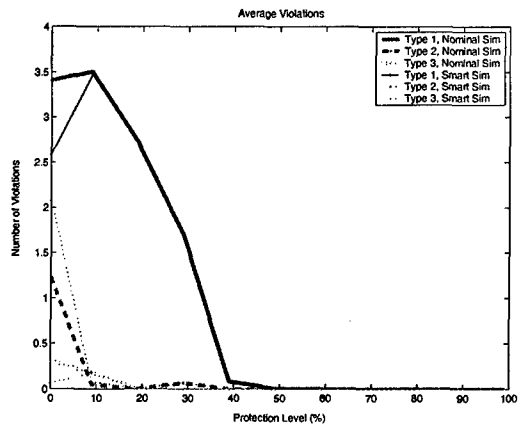


Figure 5-49: Sim and objective function values for Scenario 1

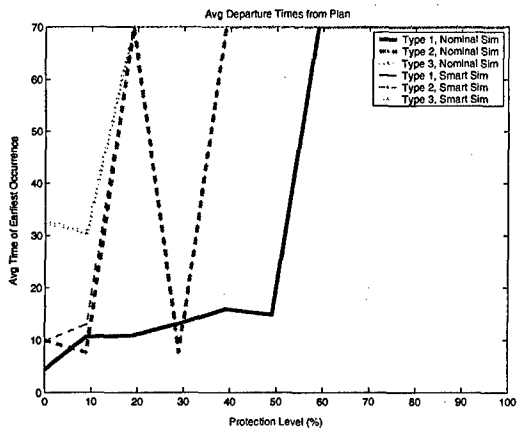
Figure 5-49 indicates that average simulation value actually increases at the 10% protection level, before it gradually decreases again. The downward slope after the 10% level is very gradual, and contains many small changes in value. A total of 8 changes in value appear in the average reported simulation value over the tested protection levels. This figure also reports that, while the initial objective function drop is not as substantial as for other scenarios, a large drop in objective function value does exist at the 40% level.

Graph (a) of Figure 5-50 reveals that a fair number of violations actually do occur in spite of the small hat values in this scenario. The violations at the unprotected level closely resemble those of the Base Scenario at this level, but decrease more rapidly. Type 1 violations decrease particularly rapidly, and almost no Type 1 violations occur at and beyond the 40% protection level. That the constraint violations diminish so quickly in the protection levels is apparent also in Graph (b), as earliest departure times reach 70 hours at levels lower than observed in other scenarios. A possible reason for the quick drop in Type 1 violations is given by the fact that, while lateness values still shadow the Type 1 violation counts, they are substantially lower than is typical. The violated Separation Constraints, then, exceed their RHS bounds by little, and can be eliminated with little added protection.

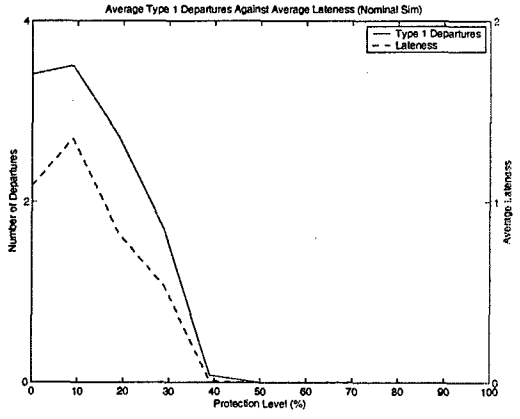
The fact that the decrease in hat values in this scenario did little to impact constraint violation in the unprotected level indicates that even small amounts of uncertainty (less than those present for Scenario 1) are sufficient for producing constraint violations, and that additional uncertainty does little to increase violation. Furthermore, the smaller levels of uncertainty in Scenario 1 actually cause protection to appear more effective in this scenario than in the Base Scenario. As noted earlier, this might be because the violations that do occur exceed their RHS limits by little, and are eliminated with a small amount of protection.



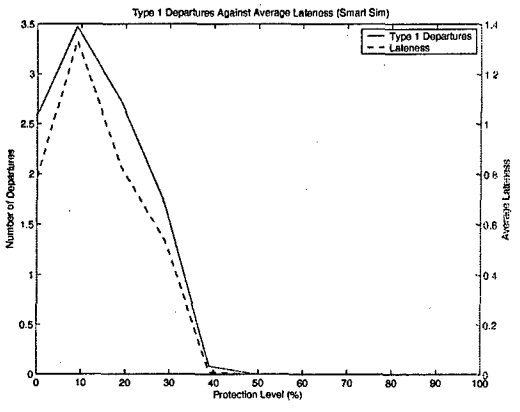
(a) Scenario 1 constraint violations



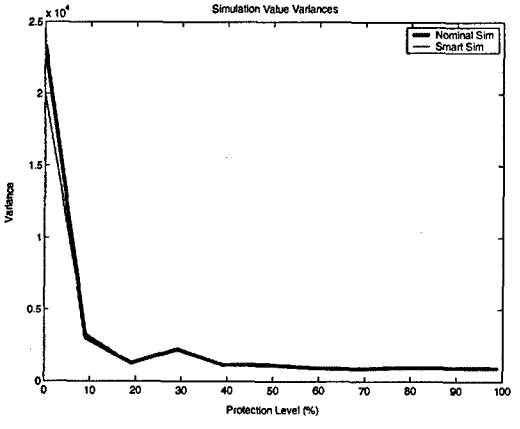
(b) Earliest departure times for Scenario 1



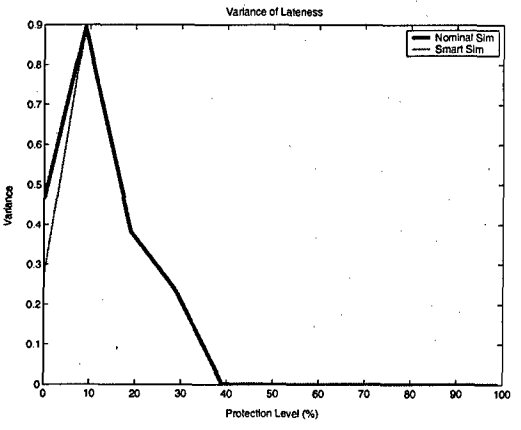
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-50: Experiment 5, Scenario 1 results

Scenario 2: In this Scenario, objective function, service time, and opportunity window hat values associated with MAUs assume a value equal to 30% of the base value, while all other hat values take on 50% of the base values. Simulation values reported for Scenario 2 (Figure 5-51) appear similar to those reported for the Base Scenario. Interestingly, while the objective function value manifests an atypically large drop at the 10% level, simulation value does as well, and both objective function and simulation values are very close at 100% protection. This may be because of the fact that the high degree of uncertainty prohibits protected solutions from achieving much. The plan at the 100% level, for instance, chooses only to perform 2 ISR tasks. According to this reasoning, then, the objective function and average simulation values do not approach one another, but rather, both values approach 0. While the drops are more substantial than those in Scenario 1, they are fewer, and only 5 changes in objective function value occur throughout the protection levels.

Graph (a) reports that, once again, constraint violation is very similar to that which is reported in the Base Scenario. The results differ from the Base Scenario in that Type 1 violations decrease monotonically and actually are eliminated before 100% protection is applied, and Type 2 and 3 violations occur longer into the protection levels. Graphs (c) and (d) indicate that lateness is higher than in Scenario 1, and closely resembles average Type 1 violations in magnitude. Graphs (e) and (f) indicate that the variances of simulation value and lateness are higher and diminish more slowly than in Scenario 1 and the Base Scenario.

The larger values for lateness and the persistence of Type 1 violations are consistent with the observation in Scenario 1 that having little lateness is associated with the improved performance of protection. Here, the greater values for lateness require that more protection be added before violations can be prevented. We now address the matter of Scenario 1 experiencing a greater number of changes in average simulation value than Scenario 2, and each of these adjusting the average value by less than the Scenario 2 changes. The Bertsimas-Sim formulation we implement uses the bounds on coefficient uncertainty when it adds protection. When smaller bounds (hat values) are present, increases in protection prompt smaller adjustments to the

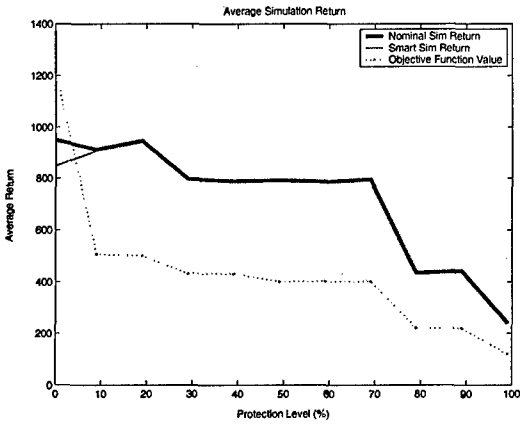
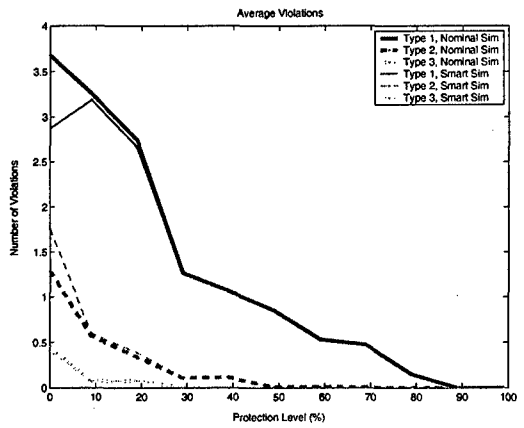


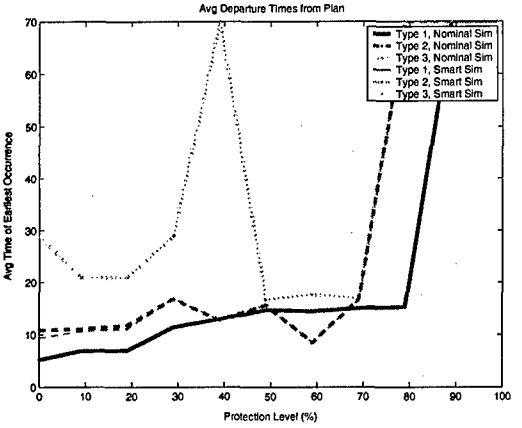
Figure 5-51: Sim and objective function values for Scenario 2

solution, which are reflected in the smaller but more frequent changes in simulation value.

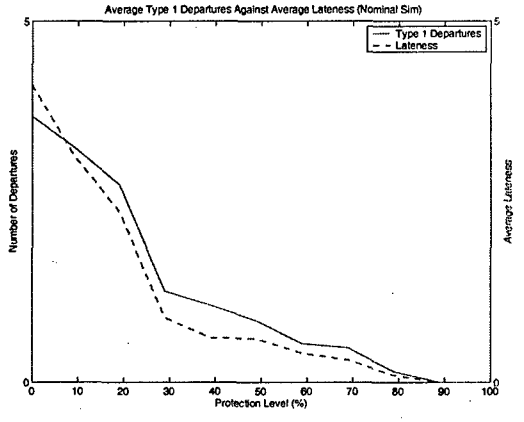
The results for constraint violation in Scenario 2 are also consistent with the observation in Scenario 1 that added uncertainty did not seem to affect constraint violation at the 0% level once uncertainty reached its levels for Scenario 1. This is somewhat contrary to our underlying beliefs for what determined constraint violation, especially when considering the fact that the solutions for the three scenarios are equivalent at the 0% level. Identical solutions for these scenarios are logical, as the scenarios differ only by their hat values, and the unprotected solution, by definition, does not consider hat values. With identical solutions, we expected that increasing uncertainty would result in increased violations and lower simulation values, which proved not to be the case. The results are similarly inconsistent with our hypothesis that added uncertainty would cause an increase in the effectiveness of protection. We decide to pursue this matter further by testing two additional scenarios: one with a degree of uncertainty smaller than that in Scenario 1, and one with hat values larger than those found in Scenario 2.



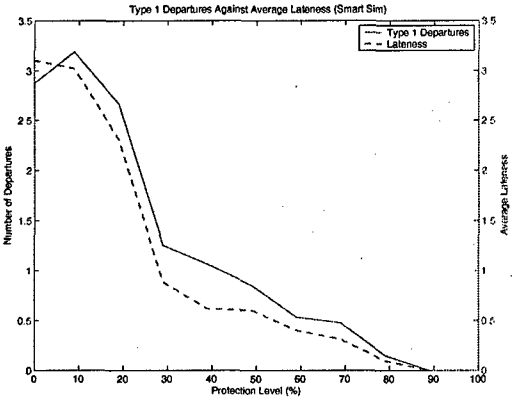
(a) Scenario 2 constraint violations



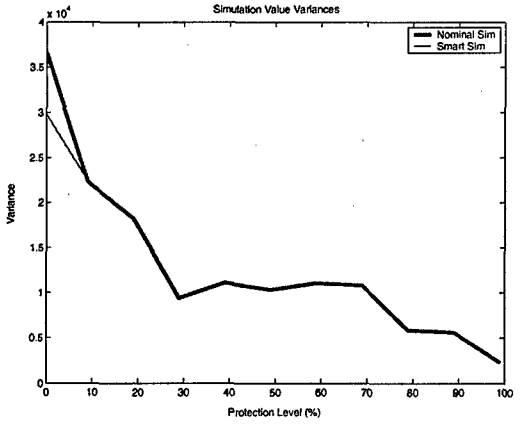
(b) Earliest departure times for Scenario 2



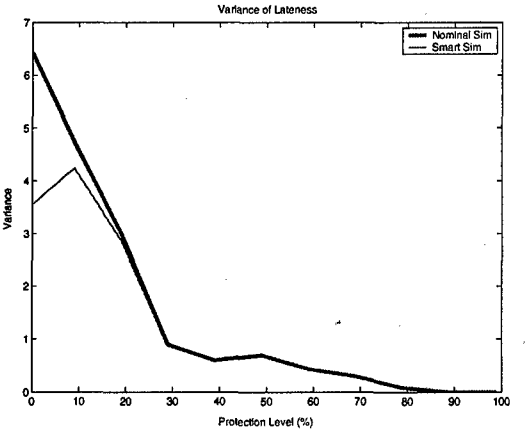
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-52: Experiment 5, Scenario 2 results

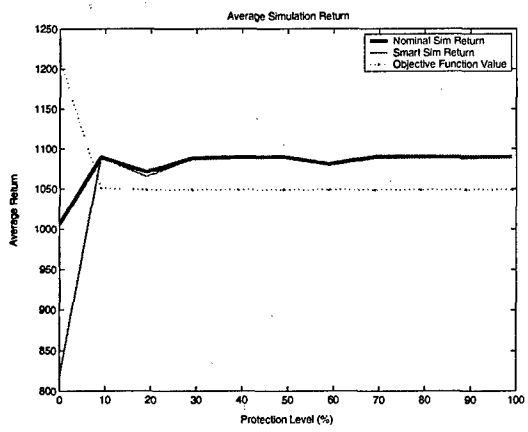
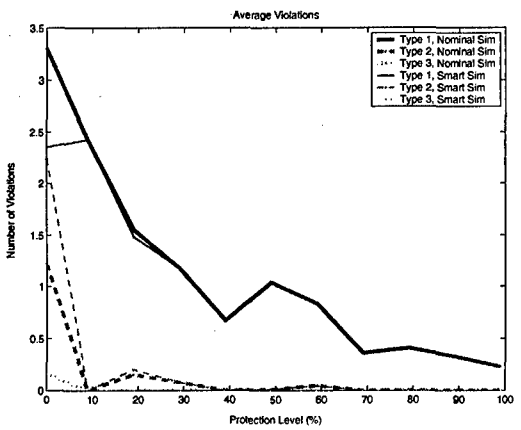


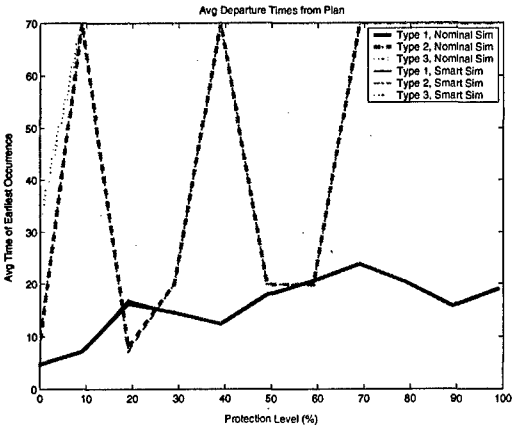
Figure 5-53: Sim and objective function values for Scenario 3

Scenario 3 In Scenario 3, the hat values associated with MAUs are equivalent to 2% of the mean values, and all other values are equivalent to 4% of their mean values. This level of the hat values represents a 75% drop from those used in Scenario 1. Results indicate that, in spite of this, the results for the unprotected solutions of the two scenarios are very similar, with Scenario 3 posting slightly fewer Type 1 and 3 violations. Figure 5-53 reports that, instead of decreasing as in Scenario 1, however, objective function value levels off once it reaches 1050 at the 10% protection level. Similarly, average simulation value does not decrease from its level at 10% protection, but remains fairly constant in the subsequent protection levels. The reported constraint violation of this scenario invalidates the belief that the rapid decrease in Scenario 1 constraint violation was caused by the small values of lateness. Graph (a) of Figure 5-54 reveals that violations persist into the protection levels for longer, and Graphs (c) and (d) reveal lateness levels that are even lower than they were for Scenario 1.

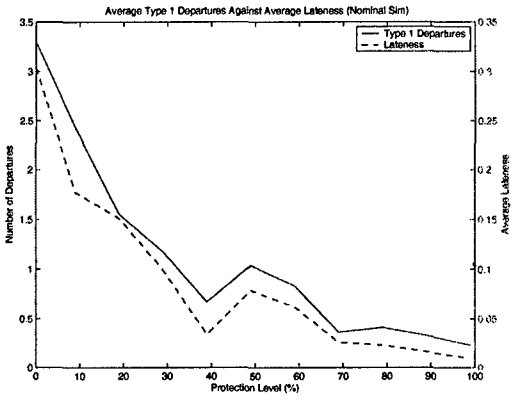
Scenario 4 In this scenario, MAU hat values are 60% of their associated mean values, while all other hat values are set to 100% of the mean values. In contrast to results of the previous scenarios, the results for this scenario report different values for the metrics at the unprotected level. While objective function value at 0% protection remains the same, simulation values are markedly lower. Both objective function and simulation values drop off rapidly, and reach 0 by 30% protection.



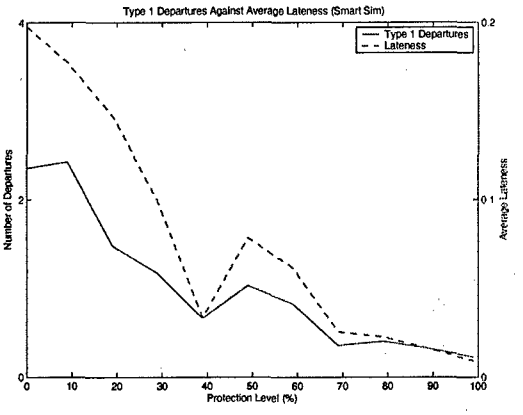
(a) Scenario 3 constraint violations



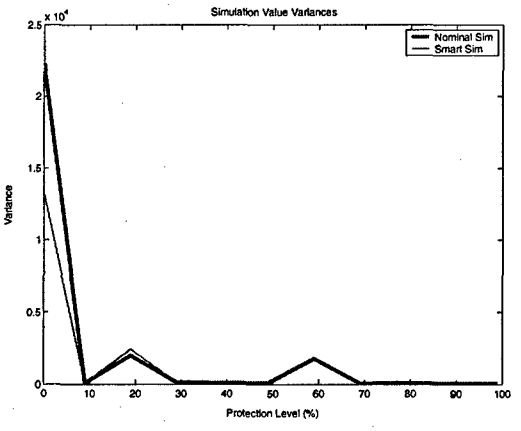
(b) Earliest departure times for Scenario 3



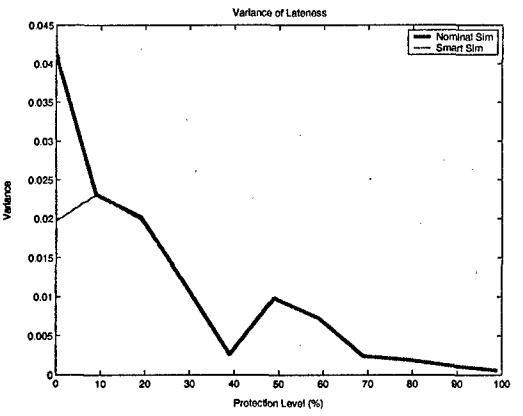
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-54: Experiment 5, Scenario 3 results

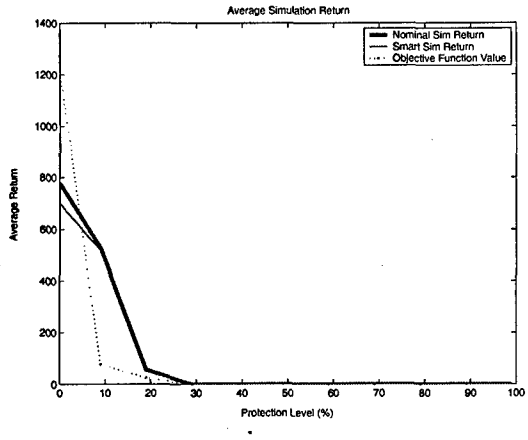


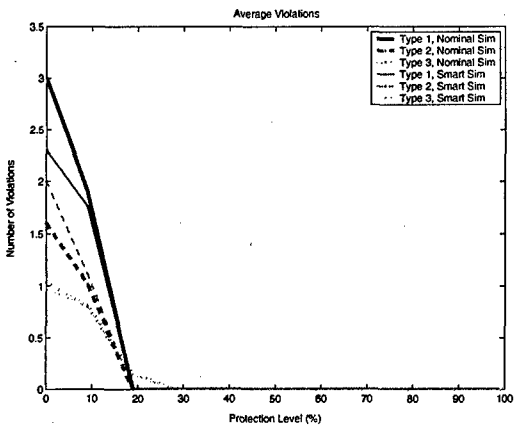
Figure 5-55: Sim and objective function values for Scenario 4

While all counts increase, the main change in constraint violation at 0% protection occurs in the Type 3 violations, which double from their value in Scenario 2. Figure 5-56 reports that, outside of simulation value, all metrics improve rapidly, and most achieve their maximum levels by 20% protection (Type 3 violation count and departure time take until 30% protection).

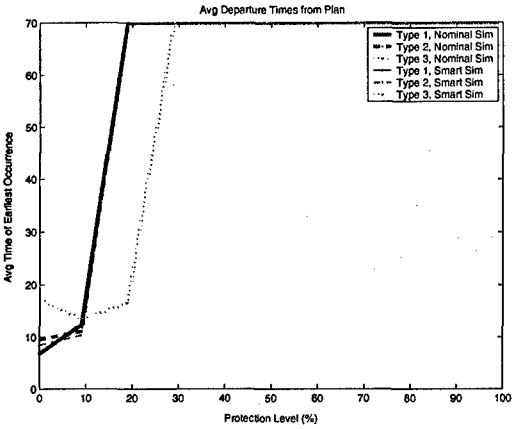
Improvement in constraint violation, departure times, lateness, and variance, is very good for Scenario 4, but it comes at the cost of earned simulation value. These metrics report such good values at 30% protection because the optimal solution here is to perform no tasks. The intermediate option, at 10% protection, gives up less simulation value, but experiences little improvement in the metrics. In this sense, scenarios with less uncertainty, such as Scenarios 1 and 3, are more favorably inclined to the use of protection, as more intermediate options exist, and a more desirable level of tradeoff can be chosen. In fact, no tradeoffs had to be made at some protection levels of these scenarios, as all metrics improved. We note, that for Scenario 4, it is possible to test smaller protection level intervals in order to find a suitable tradeoff between simulation value and the other metrics.

The lower simulation values in Scenario 4 are ostensibly a result of the increase in Type 3 violations, which occur before 20 hours and prevent the plan from performing a strike at Target 8. It is perceived, then, that the range uncertainty passed a threshhold necessary in order for range violations to occur with the UAV that visits Target 8.

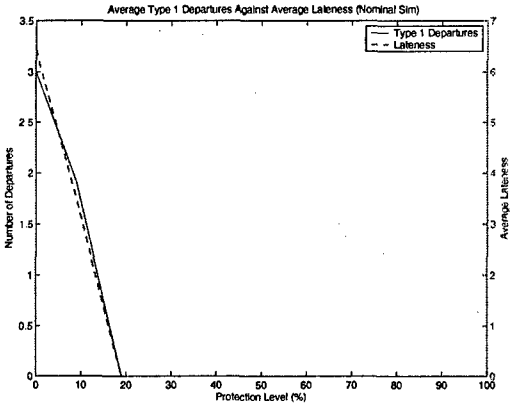
The threshhold for all of the other violations that exist at the unprotected level had already occurred by the level of uncertainty in Scenario 1, and all additional uncertainty acted to increase the amount by which the violations occurred, but not the average number of violations. Through this discussion, it appears as if the protection imparted by the Bertsimas-Sim formulation is more suitable for situations in which uncertainty is contained within smaller ranges. In these cases, as in Scenarios 1 and 3, it is possible for all metrics to improve with protection, indicating clearly that the protected solutions are favorable to the unprotected ones.



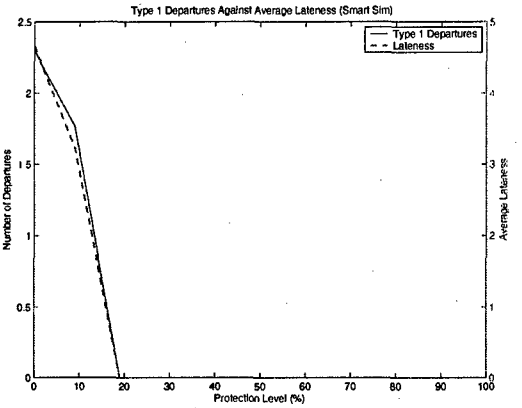
(a) Scenario 4 constraint violations



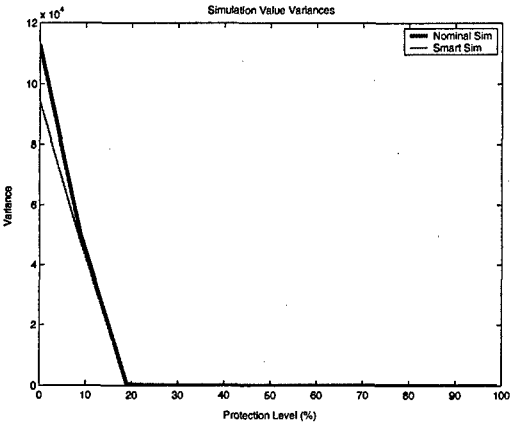
(b) Earliest departure times for Scenario 4



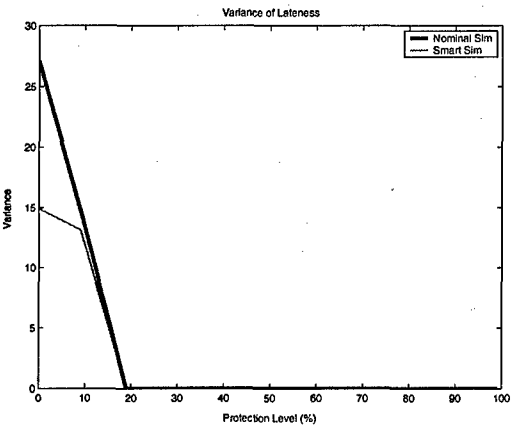
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-56: Experiment 5, Scenario 4 results

5.4.7 Experiment 6

Motivation

In the past few experiments, we observed a dependency between the constraint types. In Experiment 3, we noted that uncertainty within one set of constraints (the Separation Constraints) caused violations in another set of constraints (the Opportunity Window Constraints). In Experiment 4, we noted that manipulating the coefficients of one set of constraints increased tightness in another set, causing this set to incur more violations during the simulation. In that experiment, the reduction in travel times (which appear in the Separation Constraints) allowed plans to schedule more tasks, and prompted Opportunity Window Constraints to be tightened and consequently violated more frequently.

In Experiment 2, we noticed that metric values occasionally deteriorated as protection was added. Our conjecture was that such behavior was a result of a conflict between the protection of different constraint sets, and that added protection for one constraint set might inadvertently deteriorate the metrics associated with another constraint set. The Separation Constraints, for instance, dictate the minimum spacing between subsequent arrival times. Increasing protection for the Separation Constraints acts to increase the minimum spacing between two contiguous arrival times. If a solution is to continue scheduling the two arrivals together even after protection is added, it must either schedule the first arrival to occur sooner, the second arrival to occur later, or both. As a result of this increased spacing, the arrival times now occur closer to one of their opportunity window boundaries. The arrival times continue to tend toward these boundaries as separation protection is increased until it is no longer feasible for the two arrival times to occur after one another, and another routing must be selected. Our hypothesis concerning the behavior in Experiment 2, then, is that the protection-induced slack in one constraint adds tightness and deteriorates performance in another constraint.

In this experiment, we test the hypothesis by protecting the constraints selectively and observing any effects that doing so might have among the other unprotected

constraints. We expect that the metrics (i.e. average number of violations, departure times, and variances) that correspond to the protected constraints should improve with protection, and, if our hypothesis is true, the metrics corresponding to the remaining constraint types should deteriorate with protection.

Hypotheses

- Protection of one constraint can lead to worse performance in other constraints.

Scenarios Tested

All scenarios we test use the Base Scenario but protect different constraint sets.

1. Protect only the objective function;
2. Protect only the Separation Constraints;
3. Protect only the Time Window Constraints; and
4. Protect only the Range Constraints.

Results and Analysis

Scenario 1- Objective function protected. We first make observations from the output. As in the previous experiments, objective function value drops dramatically in the first protection level. After protection level 20%, however, it remains constant at 866.5 for the remainder of the protection levels. Because we only protect the coefficients of the objective function, we expect that the same solutions are feasible across all protection levels, and that a change of solution will occur only when the added protection causes visiting the lower-value but lower-risk Mobile Artillery Units (MAU) to garner more reward than visiting the higher-value but higher-risk leadership targets. In other words, a switch from visiting a leadership target to visiting a MAU will be made if $C_{MAU} - \hat{C}_{MAU} > C_L - \hat{C}_L$, where C_{MAU} and C_L are the mean reward values for MAUs and leadership targets, respectively, and $\hat{C}_{MAU}$ and $\hat{C}_L$ are the MAU and leadership hat values. In the Base Scenario that we test, this does not

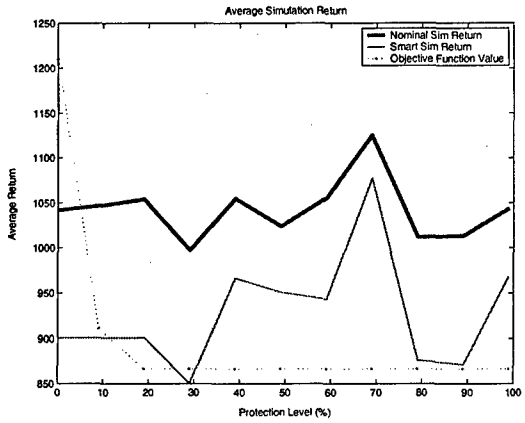


Figure 5-57: Sim and objective function values when the objective function is protected

hold for either ISR or strike tasks: $100 - 15 < 200 - 60$, and $30 - 4.5 < 120 - 36$. Because we assign hat values as a percentage of the mean value, leadership targets will always return higher objective function values, and no solution changes should exist within the protection levels. The only source for the changes in objective function value across the protection levels, then, is that of the decreasing value of visiting the leadership targets.

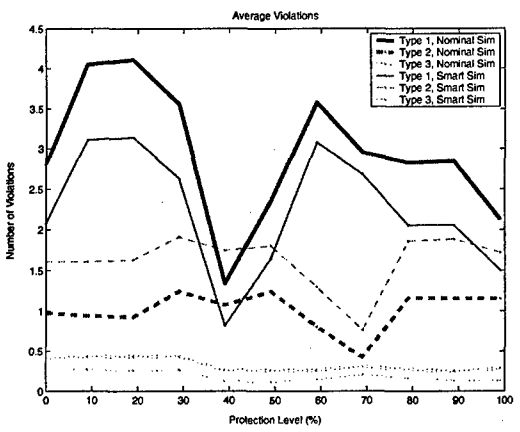
That all of these changes occur within the first 2 protection levels is logical. All solutions in this scenario call for 4 strikes and 5 ISRs to be performed. These 9 tasks represent only $\sim$ 14% of the 63 possible tasks (possible tasks $= N(K + UAV_A) = 9(4 + 3) = 63$. $\frac{9}{63} = \frac{1}{7} \approx .14$). Because only the hat values corresponding to these tasks can ever enter the objective function, the maximum protection for the objective function is reached by this protection level, and adding protection beyond the 15% level has no effect on the objective function value. Because most of our solutions to previous scenarios scheduled similar amounts of tasks, this observation holds for the previous experiments as well. Changes in objective function value in the higher protection levels of previous experiments, then, were prompted by actual changes in the solution, and not because of objective function hat values, as we had previously thought.

Figure 5-57 indicates fairly erratic simulation value results, as well as a great deal

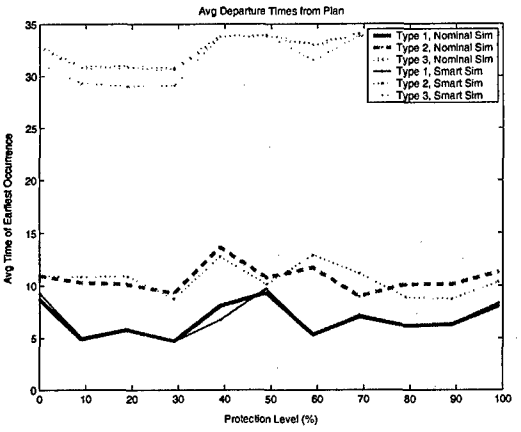
more separation between the two simulation values. After investigating the solutions, we confirm that no changes occur in the tasks that the plans call for. While they may have different arrival times and instruct the tasks to be performed at different targets, all plans call for the strike of 3 leadership targets and 1 MAU, and the ISR of 1 MAU and 4 leadership targets. Because of this, the expected objective function value generated from the tasks should be equal for all protection levels, and any observed changes in simulation value must be due to differences in the average number of Type 2 and Type 3 violations. That the two simulation values have so much spacing indicates that arrival times are close to the opportunity window and range limits, and that the Smart Sim often makes the decision to skip a target.

Graph (a) of Figure 5-58 also reveals a fair amount of separation between the lines representing the different simulations: the Nominal Sim reports more Type 1 violations, fewer Type 2 violations, and more Type 3 violations. These differences are expected. While the targets that the Smart Sim skips cause it to incur Type 2 violations, it experiences fewer Type 1 and 3 violations, because it is more likely to satisfy target Separation Constraints and less likely to encounter a range violation.

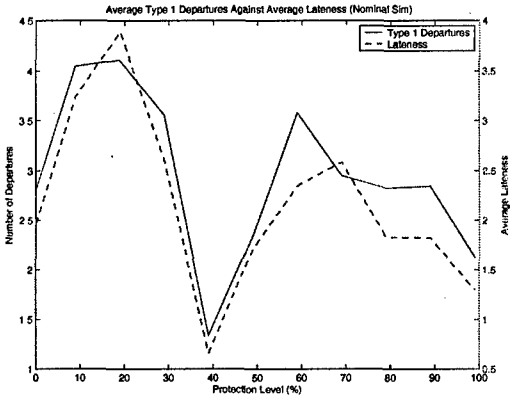
Graphs of Figure 5-58 don't point to any particular trend in any of the metrics. While Type 1 violations do change a fair amount over the course of the protection levels, they do not do so in a consistent manner. Type 2 and 3 violations remain fairly constant, as do departure times for all violation types. This is expected, as increasing protection to the objective function does nothing to the constraints. While changes in solutions do occur throughout the protection levels (the cause for the differences in average constraint violation, departure times, and variances among the solutions), they appear to be due to the solver's indifference between these various solutions, and do not appear to be a direct result of added protection.



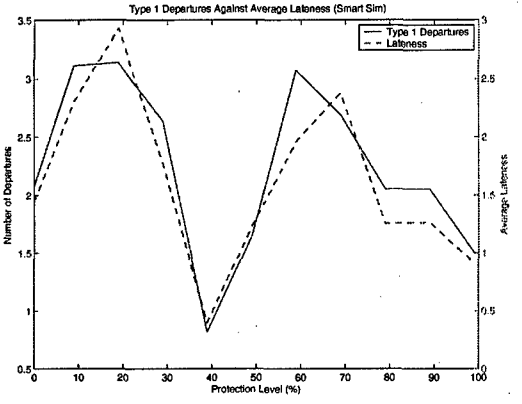
(a) Scenario 1 constraint violations



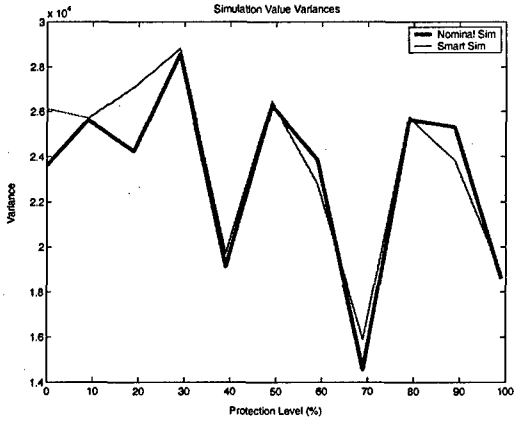
(b) Earliest departure times for Scenario 1



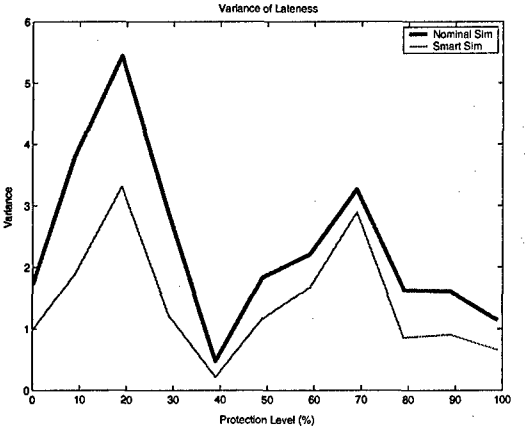
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-58: Experiment 6, Scenario 1 results

Scenario 2- Separation Constraints protected. The first observation regarding the results to Scenario 2 is that the objective function remains above both simulation values throughout the protection levels (Figure 5-59). This is expected, and is a result of the fact that the objective function is not protected in this scenario. The awards assigned to task accomplishment in this scenario are the average ones, and simulation value should not exceed them on average (and will actually be less than them when scheduled tasks are not accomplished). Also, because the objective function is not protected, the changes in the curve reflect only changes in the plans, which occur as the added protection to the Separation Constraints allows fewer tasks to be accomplished.

Interestingly, while the Nominal Sim value is substantially lower than the objective function value at the unprotected level, it closes the gap with its upper bound by the 10% level and remains close to it for the remainder of the protection levels. Violations of the Separation Constraints are not reflected in simulation value, and the elimination of the gap between the objective function and Nominal Sim values suggests that the added protection to the Separation Constraints reduced the number of Type 2 violations. This is confirmed by Graph (a) of Figure 5-60, which shows a decrease in both Type 1 and Type 2 violations. Graph (b) shows that this improvement is extended to the departure times for these two constraint types.

The improvement in the Opportunity Window Constraint violations is somewhat unexpected. The unprotected solution reveals that many of the arrival times occur very close to the opportunity window limits, generating the Type 2 violations observed at this level. Once protection is added to the separation constraints, some of these routes are no longer feasible, and protected plans contain routes that allow for arrival times further from their opportunity window limitations. In this scenario, then, we have an example where protecting one constraint can accompany an improvement in the performance of other constraints.

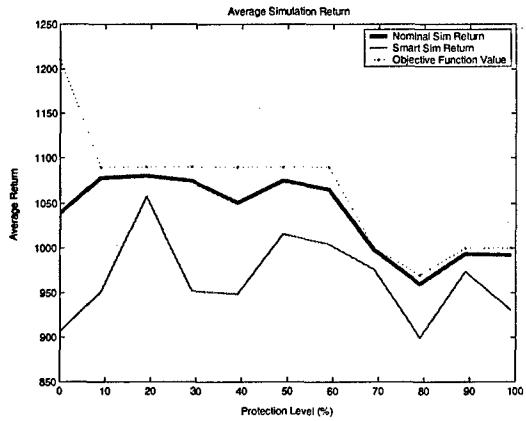
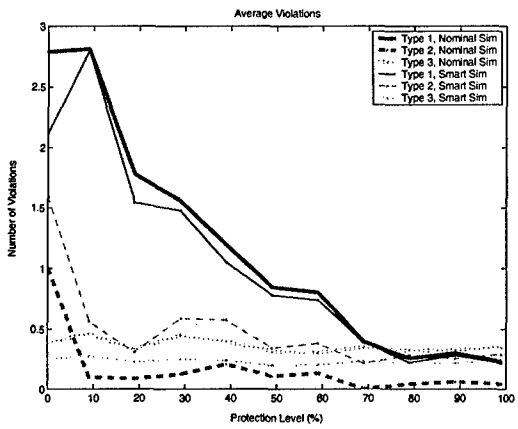
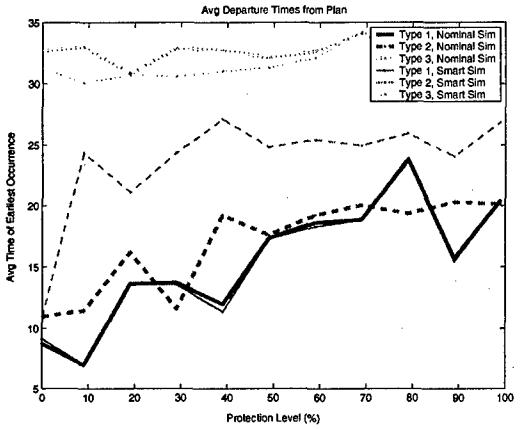


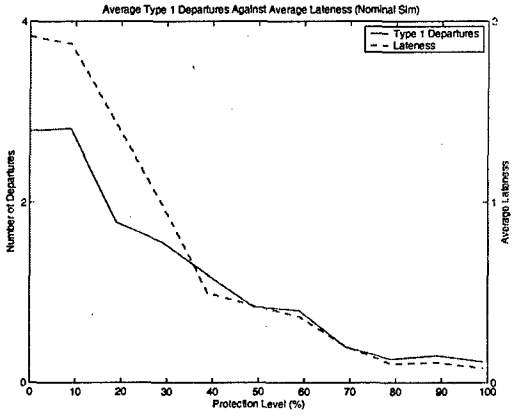
Figure 5-59: Sim and objective function values when Separation Constraints are protected



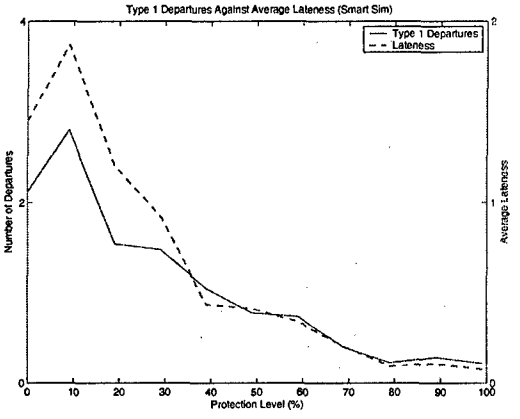
(a) Scenario 2 constraint violations



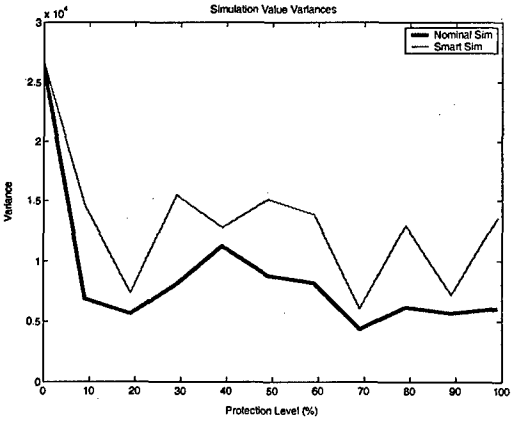
(b) Earliest departure times for Scenario 2



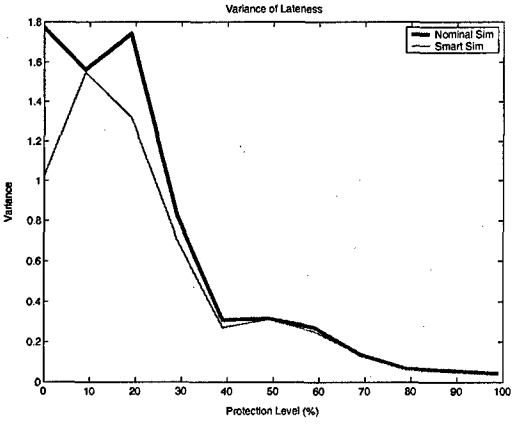
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-60: Experiment 6, Scenario 2 results

Scenario 3- Opportunity Window Constraints protected. Figure 5-61 reports the objective function and simulation values when Opportunity Windows are protected. As in the previous scenario, average simulation values remain below the objective function value. While the Nominal Sim value does move up to join the objective function value at the 10% level, it does not follow the objective function as closely as in Scenario 2. This is somewhat surprising, as we expect our protecting of the Opportunity Windows to result in fewer Type 2 violations and consequently produce a simulation value that more closely resembles the objective function. Indeed, while Graph (a) (Figure 5-62) reports improvement in Type 2 violations, the improvement is not as pronounced as in Scenario 2. Likewise, improvement in departure times occurs, but is less consistent than it was in the previous scenario. Metrics related to the other constraint types show no improvement at all.

One possible reason for the less-than-expected improvement in the opportunity window metrics is the fact that constraints governing both the start and end of the windows are protected in this scenario. The protection of each of these is in conflict with one another, as increasing protection against violation of the start of the constraint necessarily involves reducing the slack between the arrival time and the end of the opportunity window. The fact that improvement is less than expected may be in keeping with the principle set forth in the motivation of this experiment, then, as this would mean that added protection in one area comes at the detriment of another area. Because we do not track violations of the start of an opportunity window (our assumption is that the UAVs can simply loiter until the window opens if they happen to arrive too early), we decide to insert a scenario where only the opportunity window close times are protected.

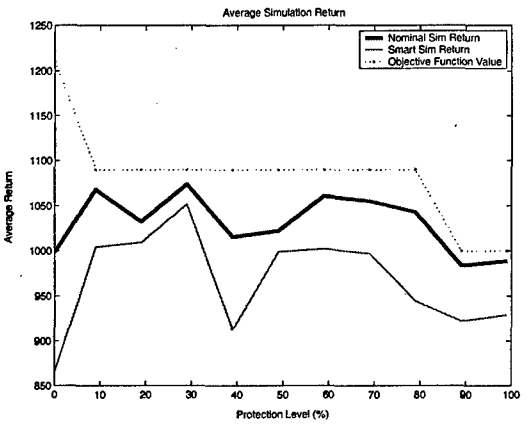
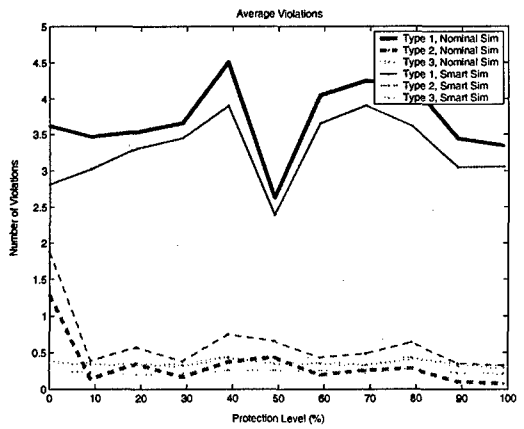
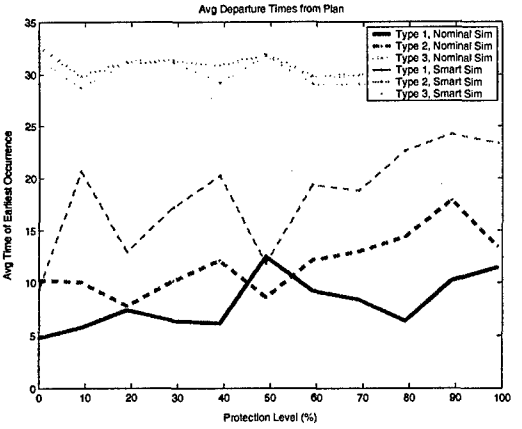


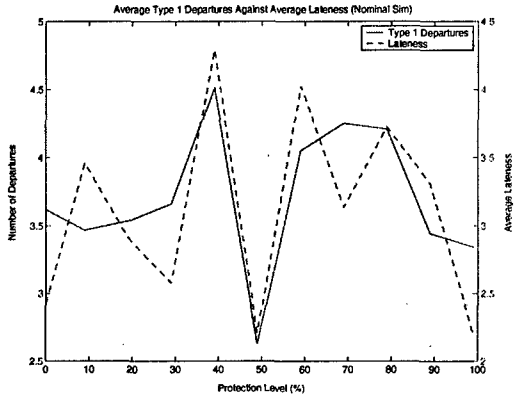
Figure 5-61: Sim and objective function values when opportunity windows are protected



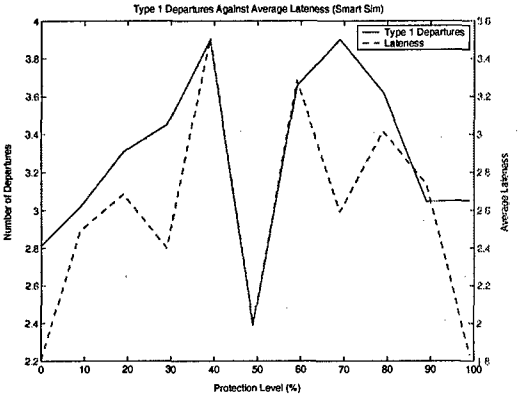
(a) Scenario 3 constraint violations



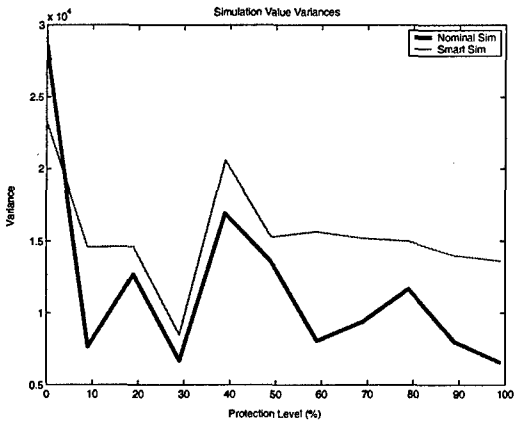
(b) Earliest departure times for Scenario 3



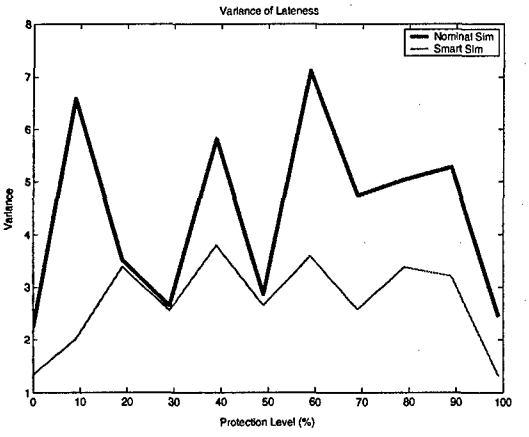
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-62: Experiment 6, Scenario 3 results

Scenario 3a- Opportunity Window Close Constraints protected. Interestingly, while the results for this scenario do show an improvement in Type 2 violations, the improvement is more gradual than in Scenario 2. Simulation value, too, improves steadily throughout the protection levels, but does not track objective function value like the results to Scenario 2. Finally, departure times for Type 2 violations show no improvement, again highlighting the fact that Bertsimas-Sim protection does not discriminate between departure times (Figures 5-63 and 5-64).

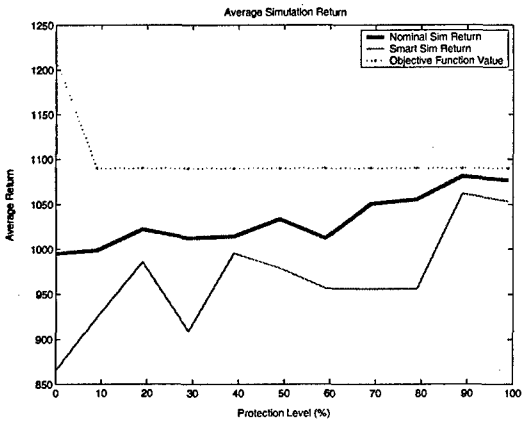
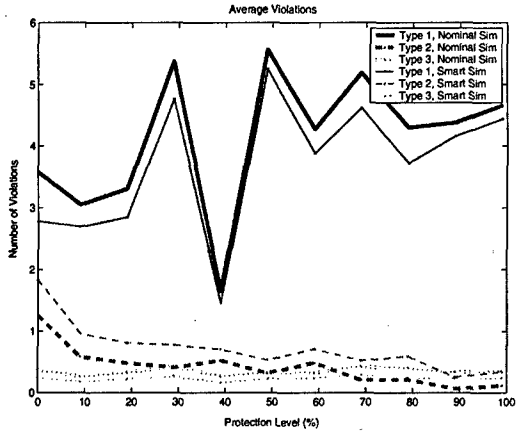
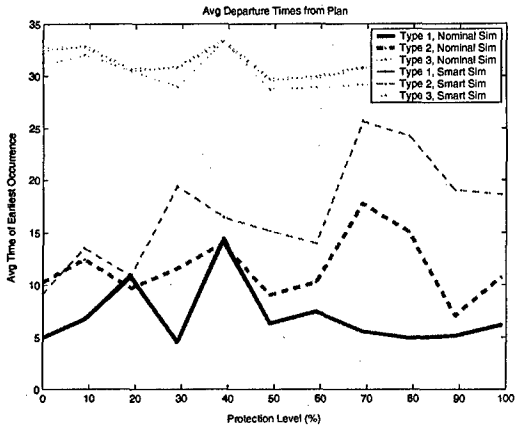


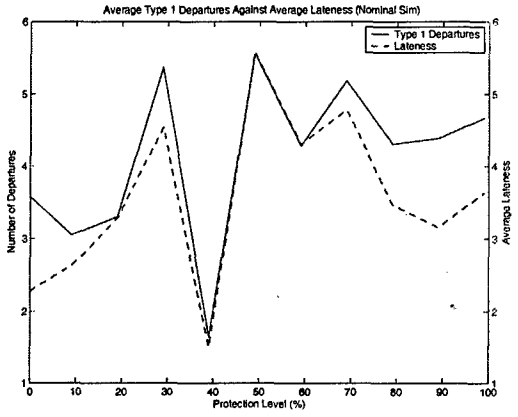
Figure 5-63: Sim and objective function values when Opportunity Window close times are protected



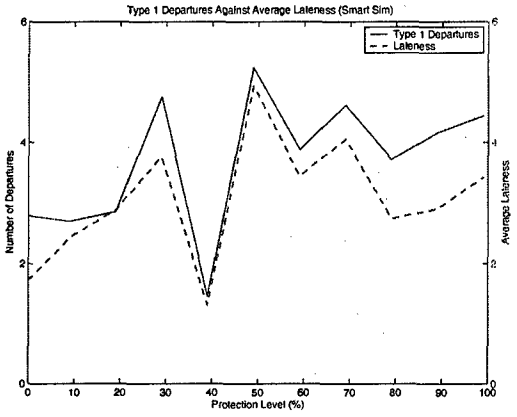
(a) Scenario 3a constraint violations



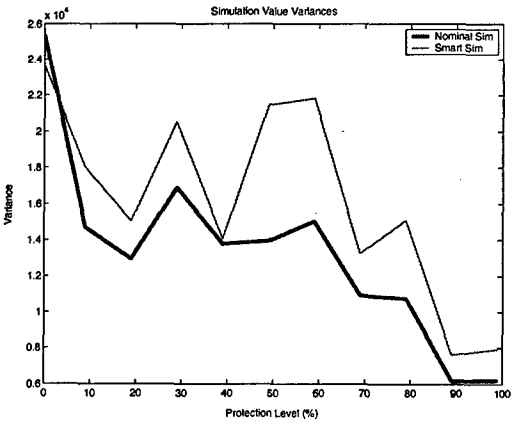
(b) Earliest departure times for Scenario 3a



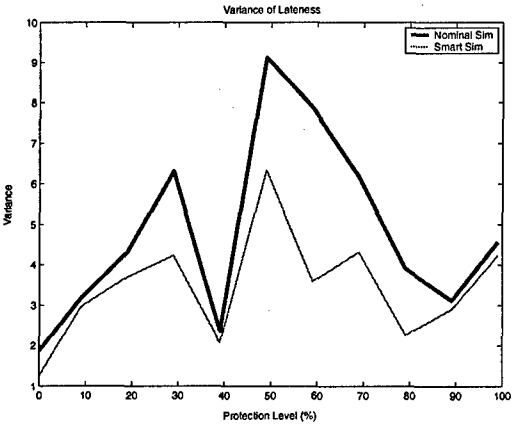
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-64: Experiment 6, Scenario 3a results

Scenario 4- Range Constraints Protected. The results to Scenario 4 (Figures 5-65 and 5-66) indicate that increasing protection to the range constraints does indeed reduce Type 3 violations. The solutions to this scenario do not perform particularly well in terms of the other metrics, though, and simulation values keep their distance from the objective function, Type 1 and 2 violations remain at relatively high levels, and the variances show no improvement. Even though the other metrics show no improvement, neither do they appear to worsen, and the results to this scenario, like those of the previous scenarios in this experiment, fail to reveal behavior consistent with our hypothesis.

As noted, the results to this experiment suggest that our hypothesis that protecting one constraint set would cause added violation in another constraint set is not correct. There was no evidence that adding protection in one constraint type consistently increased violations in any other constraint type, and violations in the other constraint types even dropped when Separation Constraints were protected in Scenario 2.

These results demonstrate that added protection in one constraint is limited in its ability to negatively impact other constraints. Though it still seems evident that constraints are interrelated, and while our results do not rule out the possibility that adding protection in one constraint might affect tightness in another, it still stands that the decision variables of feasible solutions cannot push constraints beyond their RHS limitations. Regardless of the influence protection of one constraint has on the metrics of another, feasible solutions must still satisfy the requirements of all constraints. The only influence we might observe, then, would come in the way of removing slack in the affected constraints. This experiment also revealed, however, that the current formulation is indifferent between solutions with different amounts of slack in the constraints. It cannot be assumed, therefore, that adding protection to one constraint will necessarily result in a reduction in slack for the other constraints, as the solver might arbitrarily select a solution with more slack in the constraints, even after protection is added to the one constraint. This type of arbitrary behavior is consistent with the results, which often appeared to move randomly as protection

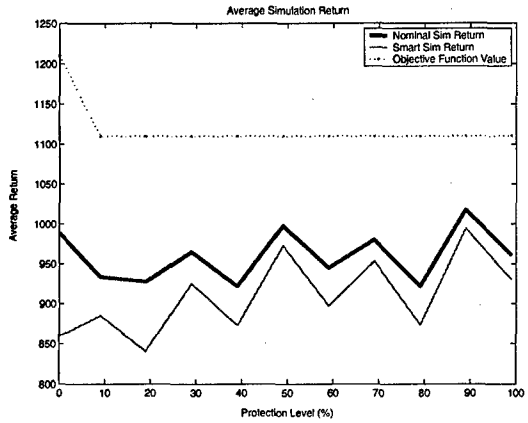
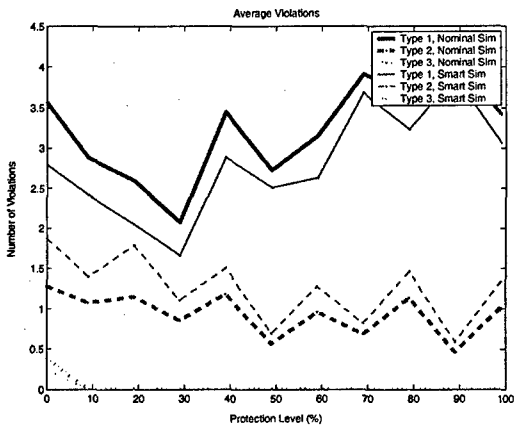


Figure 5-65: Sim and objective function values when the Range Constraint is protected

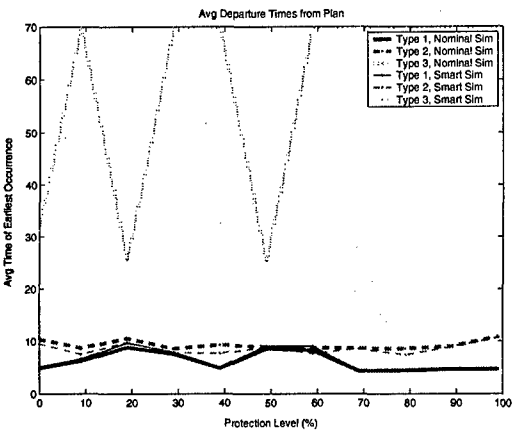
was increased.

The indifference between solutions with varying amounts of slack would also account for the often erratic behavior of the output graphs in all previous experiments, and would explain the instances when adding protection deteriorated the metrics against our expectations.

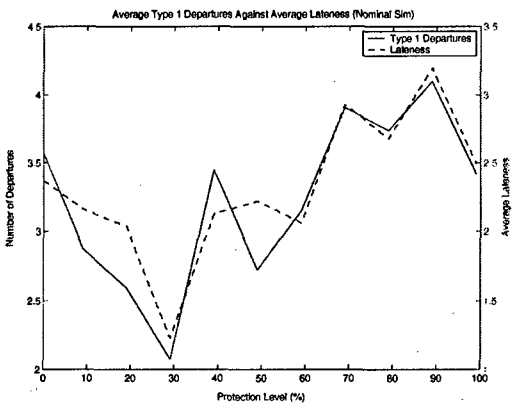
According to our discussion in the secton on metrics, solutions with different amounts of slack in the constraints are not identical. If a solution with more slack experiences fewer constraint violations and improved metrics, on average, it is a better solution. The formulation, then, should not be indifferent between these solutions, but should discriminate between solutions with different amounts of slack.



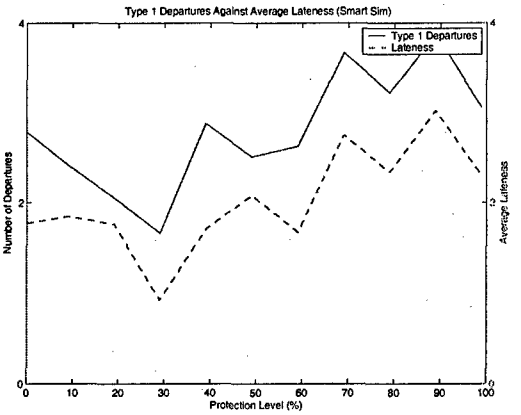
(a) Scenario 4 constraint violations



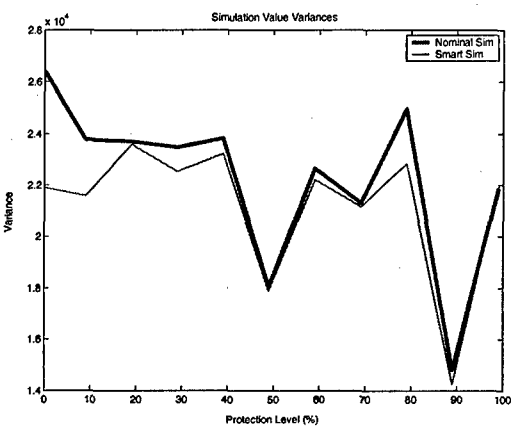
(b) Earliest departure times for Scenario 4



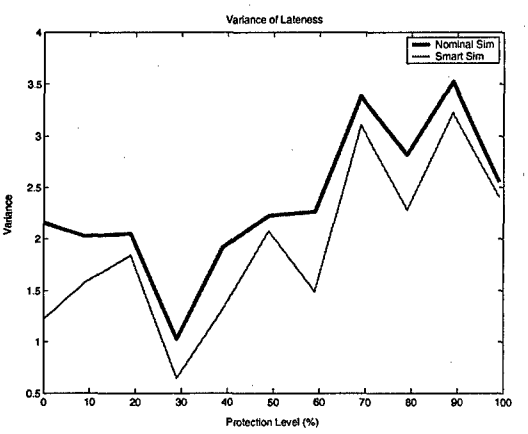
(c) Lateness statistic measured for the Nominal Sim



(d) Lateness statistic measured for the Smart Sim



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-66: Experiment 6, Scenario 4 results

5.4.8 Experiment 7

Motivation

In the last, experiment we noted that the inconsistent behavior noted in the previous experiments were not caused by dependencies among the constraints, and surmised that they were more likely due to an indifference between solutions that were all feasible and had identical objective function values, but were not identical with respect to the metrics. In particular, the formulation is indifferent to solutions that possess different levels of slack, and therefore are expected to perform differently in terms of the metrics. This experiment tests a formulation that seeks to rectify this. The new formulation updates the previous one with Lagrangian elements in the objective function. It maintains the constraints of the old, but mirrors them in the objective function in a way that favors solutions with more slack. In order to keep from overly complicating the objective function, we limit our modification to include only the constraints dealing with the opportunity window close times, as we feel that the solutions would be most profited from slack in these. The modified formulation adds the following line to the objective function:

$$+ \sum_{i \in N, k \in UAV_2} (B_i ISR_{k,i} - w_{ISR,k,i}) + \sum_{i \in N, k \in UAV_1} (B_i ISR_{k,i} + B_i STR_{k,i} - w_{ISR,k,i} - w_{STR,k,i}) \quad (5.1)$$

We expect that the protected solutions to the modified formulation will improve over the original, with respect to the metrics. We also expect improvement in the metrics to be more consistent, and that they will be monotonic with respect to protection.

Hypotheses

- Protected solutions to the modified formulation perform better than those of the original formulation, with respect to the metrics.
- Under the modified formulation, solution metrics will be monotonically improv-

ing with respect to protection.

Scenarios Tested

1. Base Scenario.

Results and Analysis

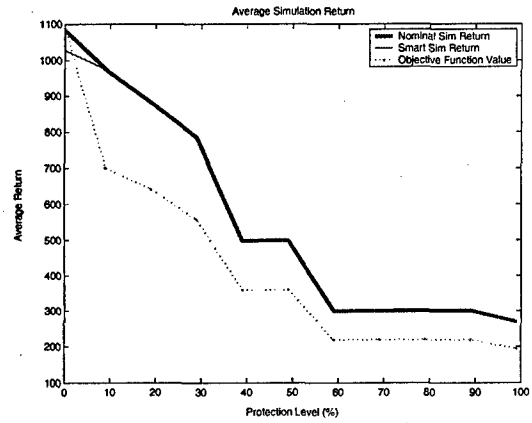


Figure 5-67: Sim and objective function values for modified formulation, under the original weighting

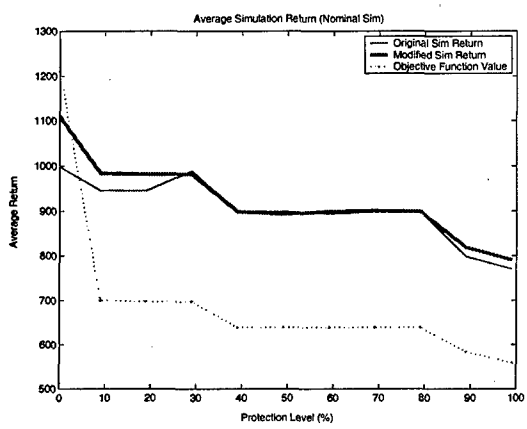


Figure 5-68: Sim and objective function values for both original and modified formulations, under the new weighting

Our intention in the modification was that the formulation would provide solutions that retain the same objective function value at each protection level, but

improve on simulation value by adding slack to the Opportunity Window Close-Time Constraints. As Figure 5-67 indicates, however, the solutions to the modified formulation post different objective function values from those reported for the original formulation. We attributed this change in objective function value to the weighting given to the Lagrangian additions to the objective function. Because of the relatively heavy weighting of the Lagrangian elements that are added, the modified formulation is able to achieve higher objective function values than the original formulation is. This added value, then, issues from the Lagrangian, and is not necessarily a result of increased task accomplishment. As Figure 5-67 reports, the increase in objective function values is actually accompanied by lower simulation values. While higher objective function values are achieved, the benefit is not translated into simulation value, as the simulations do not directly reward the slack represented by the Lagrangians (slack can only benefit the simulations if it reduces constraint violation). The added objective function value actually deteriorates simulation value, and the weighting generates instances where adding slack is preferable to adding tasks. While the added slack decreases the chances for constraint violation, it does not increase the possible reward like adding tasks to the plan does. We alter the formulation by multiplying the Lagrangian elements by a weighting of $\frac{1}{10}$ and rerun the simulation.

In order to provide a better comparison between the results of the original and modified formulations, we include graphs where the results of the two are superimposed. Results to the modified simulation are indicated with a thick line, while those of the original formulation are designated with a thin line. As expected, the new weightings eliminate the disparity between objective function values, and the two are now virtually identical. In addition to this, Figure 5-68 reveals that the simulation values of the modified formulation show a definite improvement over those of the original formulation, and Nominal Sim value for the modified formulation appears to be greater than or equal to the original Nominal Sim value at all points. The values are also much more consistent, and shadow the objective function. Because of the similarity of the Nominal and Smart Sims in this scenario, and because of the diminished attention received by the Smart Sim through the course of the testing, we

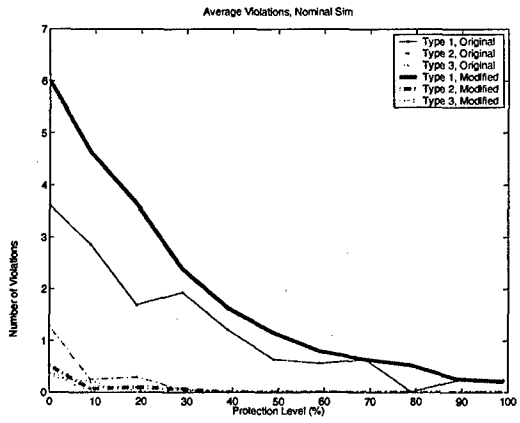
choose to report only Nominal Sim results.

While simulation value displays an improvement, Graph (a) of Figure 5-69 reports that a tradeoff is made by the new formulation, and that not all metrics improve. In particular, while the Graph reports a definite improvement in Type 2 violations (which are reduced by more than half for the first few protection levels), Type 1 violations experience a significant increase (greater than 30% at the 0% protection level). The differences between Type 1 and 2 constraint violations diminish as protection is increased. The Type 3 violations of both formulations do not differ much throughout the protection levels.

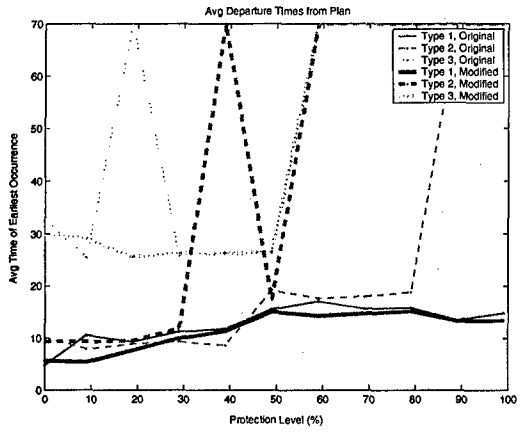
Departure times, on Graph (b), change little outside of an improvement in Type 2 departures that comes from Type 2 violations being eliminated earlier in the protection levels. Graph (c) reports the values of the lateness statistic in the original formulation, while (d) reports those of the modified formulation. The two again indicate the concession made by the modified formulation, as lateness increases significantly in it. Graphs (e) and (f) report a decrease in the variance of simulation value in the new formulation, and an increase in the variance of lateness.

The tradeoffs between metrics that the modified simulation makes exposes the dependencies between the constraints (and, in particular, the Separation and Opportunity Window Constraints) that we searched for in the previous experiment. These dependencies exist not in the constraint bounds, which must always be satisfied, but in the amount of slack given to the constraints. The Lagrangian elements of the modified formulation add slack to the Opportunity Window Constraints, and consequently remove it from the Separation Constraints. This results in fewer Type 2 violations and improved opportunity window metrics, but more Type 1 violations and worse Separation Constraint metrics. As the graphs indicate, the slack that is induced upon all constraints by the added protection displaces the slack that is the emphasis of the Lagrangian modification, and the metrics of the two formulations converge as protection increases. Graph (b) reveals once again that the formulation, even in its modified state, does not discriminate between violations on the basis of time.

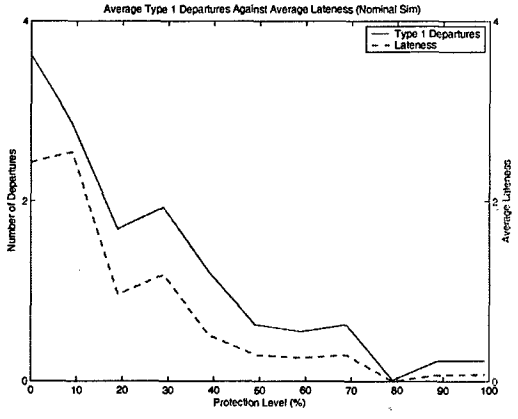
The results of this experiment are consistent with our hypotheses. The metrics report values that are more consistent, and changes in the metrics are monotonic for the most part. The modification does not improve all metrics, however, and a tradeoff exists. As the results indicate, the modification leads to an improvement in the metrics of the constraint inserted into the objective function, but can deteriorate those that correspond to the other constraints. We feel that the modification presented in this experiment reveals that benefits can be reaped from introducing Lagrangian representations of constraints in the objective function. In particular, this modification might be useful when some constraints are deemed more important than others. Introducing these constraints in the objective function increases their slack and improves their metrics, while still ensuring the required degree of protection in all other constraints.



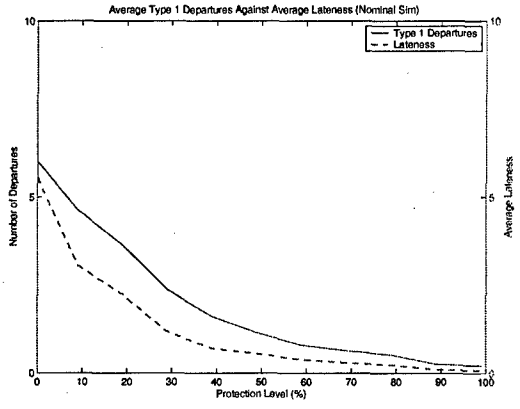
(a) Scenario 1 constraint violations



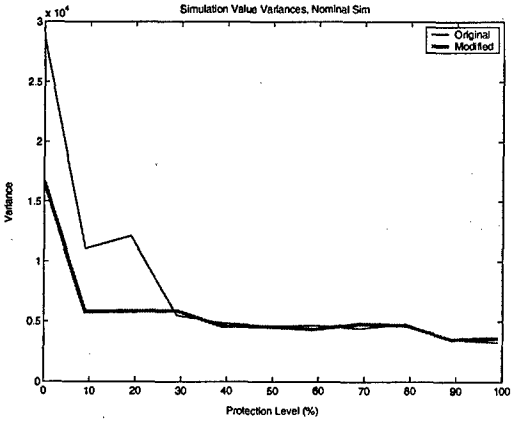
(b) Earliest departure times for Scenario 1



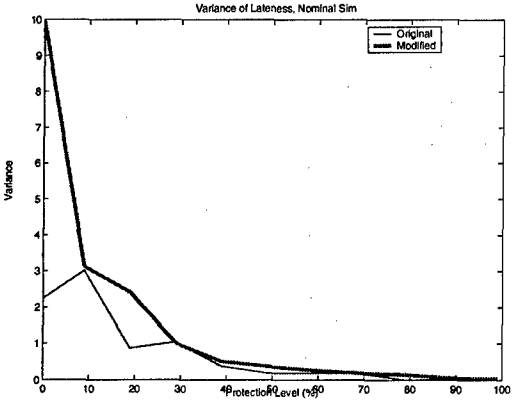
(c) Nominal Sim lateness statistic, original formulation



(d) Nominal Sim lateness statistic, modified formulation



(e) Variances of sim value



(f) Variance of the lateness statistic

Figure 5-69: Experiment 7, Scenario 1 results

5.5 Summary

We summarize this chapter by reiterating the significant findings of the experiments.

- Unprotected solutions of our UAVMPP formulation already possess a degree of robustness. Less constraint violation was encountered throughout the testing than was originally anticipated. We attribute this to the fact that many of our decision variables are binary. This requirement of the decision variables often causes them to be removed from the boundary of the feasible region in optimal solutions (see Section 4.1 for an explanation of the feasible region). As mentioned in Chapter 4, while boundary solutions might be more optimal in terms of objective function value, they are also more susceptible to uncertainty and constraint violation, and all Robust Optimization methods operate on the principle of protecting by selecting solutions that are interior to the boundary.

- Base Hypothesis 1 (see Table 5.4) was generally consistent with the results of the testing, and violations did tend to occur later as protection was increased. This, however, was not always the case, and departure times occasionally occurred sooner once protection was added. This observation made evident the fact that the formulation, in its present form, does not weigh violations on the basis of time, and did not protect against violations that occurred earlier in the operating horizon more than it did violations that occurred later. The observed improvement in departure times among Type 2 and 3 violations, in fact, was the biproduct of their decreased violation, and did not occur because the later times were specifically selected by the formulation. Once violations were eliminated for a particular constraint type, the corresponding departure time statistic naturally tended to the maximum time of 70 hours.

  We also noted that the departure time statistic does not always give a good indication of improvement in the solutions. For scenarios in which the target opportunity window close times were identical, for instance, there was no possibility for variation in Type 2 violation departure times, as these violations

Base Hypotheses
1. Increasing protection will cause the average plan to be executable for longer periods of time before experiencing a violation
2. Increasing protection causes plans to experience fewer constraint violations on average
3. Increasing protection causes violated constraints to be violated by less, on average
4. Increasing protection increases the expected plan value to the user
5. Increasing protection causes the expected return (objective function value) to more closely match the achieved (realized) value
6. Average variance on all metrics decreases as protection is increased

Table 5.4: Base Hypotheses

by definition only occur at the close of the opportunity windows. In addition, because of the manner in which we recorded departure times, it was possible for events that were not statistically significant to heavily impact the metric. One instance of a constraint violation out of the 1000 iterations was adequate to move the departure time for that constraint type.

- In general, Base Hypothesis 2 was found to be true, and increasing the protection did result in reduced constraint violation. As with departure times, however, violations did not always monotonically decrease with protection. In such cases, we surmise that the slack already existent in the solutions tended to mask the effects of protection, and that a less-protected solution that possessed slack could experience fewer violations than a more-protected solution with no slack.

- Base Hypothesis 3 was also true in most cases. The lateness metric, which was our representation for the amount by which constraints were violated, tended to follow the Type 1 violation count, and decreased as protection increased. This did not always hold, however, and lateness occassionally increased with

protection. We believe that such behavior was due, in part, to a transfer that existed between Type 1 and 2 constraints. Increased protection averted Type 2 violations, but might have increased Type 1 violations in doing so, as the arrival times that occurred within the target opportunity windows were now subject to Type 1 violations. Such an increase in Type 1 violations would account for the increase in average lateness.

- Base Hypothesis 4, which stated that expected simulation value would increase with protection, was incorrect in most scenarios. In general, expected simulation value decreased with protection. There were, however, a few exceptions to this rule, and a few scenarios experienced increases in simulation value in the first few protection levels. An example of this was Scenario 5 of Experiment 2, which was one of the few occasions in which all metrics improved with protection. In the few instances where simulation value appeared to improve with protection, this happened within the first few protection levels, before decreasing again. In general, it seemed that simulation value increased when unprotected solutions were aggressive in their UAV utilization, and missed many of their scheduled tasks as a result. Adding protection produced more conservative solutions that accomplished a larger fraction of their planned tasks and achieved higher values as a result.

- Base Hypothesis 5 was incorrect. Instead of approaching one another, objective function and simulation values diverged as protection was increased. This was a result of the behavior of the objective function, as the Bertsimas-Sim method of protection did not appear to produce particularly meaningful objective function values. Protecting the objective function did occassionally affect plans, and generally favored low-risk but low-reward tasks over the high-risk and high-reward tasks. The protected objective function values were overly pessimistic with regard to task reward, however, and objective function value generally sank well below simulation value within the first 10% of protection. This divergence between the objective function and simulation values invali-

dated Base Hypothesis 5, and also caused the objective function to cease from being useful as an estimate of anticipated reward.

- Base Hypothesis 6 appeared to hold, and the variances of the metrics, on average, decreased with protection. For metrics such as the constraint violation count, not only did the metric averages improve with protection, but values close to these averages were more likely to be attained. This is quite significant, as solutions with better metrics on average might be less desirable if they are surrounded by large variances and much uncertainty. This argument is identical to the one that prompted us to incorporate robustness into our formulation to begin with.

- Our secondary hypotheses were sometimes true, but tended to be overly simplistic. We thought that constraints that were more restrictive and more difficult to satisfy would have solutions with less slack and increased constraint violation. This was sometimes the case, but the presence of restrictive constraints also tended to reduce the number of tasks that were scheduled, thereby increasing slack. Relaxed and less restrictive constraints, on the other hand, sometimes produced plans that were more ambitious and scheduled more tasks. Such plans had less slack in the constraints and were subject to increased violation.

- The Smart Sim did not perform as well as expected. We anticipated that the Smart Sim's decisions to avoid unattainable targets would cause it to achieve higher value and encounter fewer violations. Instead, these decisions resulted in more Type 1 and 2 violations and decreased simulation value, with little benefit in the way of fewer Type 3 violations. This poor performance was partially because few Type 3 violations were encountered in the Nominal Sim to begin with, and partially because the Smart Sim sometimes chose to avoid tasks that it could have accomplished.

- We observed a dependency between constraint types. Uncertainty within one set of constraints prompted violations in another set of constraints. Manipulating

the coefficients of one set of constraints had a similar impact on violations among the other constraints.

While protecting one set of constraints did appear to impact violation in other sets, it was limited in its ability to do so, and could only affect the slack present in the other constraints. If the formulation were indifferent regarding slack, this influence would be masked by the arbitrary selection of slack.

- Results to the experiments suggested that protection should be specified on a constraint by constraint basis. Most of our experiments specified protection in a uniform manner and as a percent of the uncertain coefficients in the constraints. In practice, however, some constraints are less likely to have a large percent of their uncertain coefficients activated in solutions. In the case of our formulation, because solutions typically did not activate a large percent of the possible uncertain coefficients in the objective function, maximum protection for the objective function was achieved early in the protection levels. Tailoring the protection on a constraint by constraint basis might result in a better mix of protection options.

- Our original formulation was indifferent between solutions with different levels of slack, even though these differences in slack impacted the metrics. In the last experiment, we demonstrated a way of modifying the formulation so that it would disciminate between solutions on the basis of slack. Such a modification resulted in improved performance in the constraints for which slack was valued, and also produced a more consistent response in performance when protection was added. This method was also useful in discriminating between constraints, and could be utilized when some constraints are deemed more important than others.

In the next chapter, we present concluding thoughts on the inclusion of protection in the UAVMPP, on the formulation itself, and on using the Bertsimas-Sim method of protection. We will provide suggestions for modifications that might be made to the UAVMPP formulation and to the Bertsimas-Sim formulation.

Chapter 6

Conclusion

We devote this chapter to general comments concerning the test results, our formulation, and the Bertsimas-Sim method of incorporating uncertainty. In it, we present our thoughts on the usefulness of protection in the formulation, and make suggestions on when and how it should be used. We make recommendations on how the formulation and the delivery of protection might be improved. Finally, we provide recommendations for future work in the field.

6.1 Usefulness of Protection

We first draw some conclusions regarding the usefulness of protection in the testing. In general, we feel that something was certainly gained by protecting the solutions. Solutions that received protection generally exhibited improvements in many of the metrics as a result. The improvement was not as inclusive as we originally expected it would be, however. While constraint violation and metric variances improved, they generally did so at the cost of expected simulation value. Only in a few scenarios did protection result in the simultaneous improvement of all metrics. Such inclusive improvement resulted when optimal solutions called for heavy UAV utilization. In such cases, the nominal solution was infeasible under many of the data realizations, and protected solutions, in spite of being more conservative, actually generated more value on average.

In some scenarios, the protection levels we tested provided ranges from which to choose the desired level of tradeoff, but in others the tradeoff resembled a zero-sum game, where protection and objective function value were mutually exclusive. This was the case when the coefficients were surrounded by sizeable uncertainty ranges, in which case no alternatives were likely to produce a favorable result.

Improvement in the metrics was often inconsistent: adding protection would improve a metric, but further protection would have the opposite effect, and so forth. This behavior resulted from the formulation's indifference between solutions with different levels of slack. The inconsistency appeared to be corrected in the modified formulation in Experiment 7.

6.2 The Effectiveness of the Bertsimas-Sim formulation

We now discuss our views on the Bertsimas-Sim formulation as a means of delivering protection. We feel that the formulation is a novel way of implementing a graduated protection that maintains the linearity and computational complexity of the underlying problem formulation. User defined protection adds flexibility to the formulation, as solutions can be tailored to the needs of the user. Because added protection usually resulted in tradeoffs among the metrics, it was unlikely that either extreme of protection (no protection or 100% protection) would be optimal to the user, and the option of choosing intermediate levels was thus quite advantageous.

While the Bertsimas-Sim formulation maintains the linearity of the original formulation, it increases the size of the problem in terms of constraints and variables, and this might affect tractability in larger problems. While we, in our testing, did encounter problems that were intractable (and thus were compelled to limit the size of the problems that we dealt with), we did not ascertain how much of this was due to added complexity from the Bertsimas-Sim formulation.

Some of this added complexity appeared to be unnecessary. As we have pointed

out, the ρ variables can be removed from constraints that have only one uncertain coefficient. In such cases, however, the Bertsimas-Sim formulation performs the same function that multiplying the uncertain coefficient by a moderating value would. Doing so would considerably simplify the formulation, as the added Bertsimas-Sim constraint and variables could be removed. Because our formulation contains many such constraints, we were able to avoid a fair amount of added structure by making this adjustment.

Do the added features of the Bertsimas-Sim formulation warrant the complexity added by the formulation? Simpler methods of protection do in fact exist. Protection can be imparted into a formulation by adjusting its coefficients to make the constraints more restrictive. We could, for instance, plan off of a required leadership target service time of 30 minutes instead of 20 minutes. This method is simpler, as it only affects the data inputs, and does not alter the formulation itself. It can still provide protection that is graduated, as the user could arbitrarily increase or decrease the coefficients. If the same degree of protection with the same tradeoff with optimality could be gained from using this decidedly simpler technique, there would be little argument for using the Bertsimas-Sim method. This research, however, does not provide any indication that this is the case, and solving the scenarios with the simpler model would be an interesting extension to the research.

We also question the use of the bounds of coefficient uncertainty ranges to protect them. Might a more favorable tradeoff be obtained through protecting *all* uncertain coefficients by a little rather than a *few* coeffients by their maximum amount? We question whether it is ever efficient to protect coefficients by the full amount. If it is primarily measurement error that we desire protection from, it is conceivable that the true distribution of uncertainty is Gaussian. In this case, we protect against an incrementally smaller proportion of the realizations with each step we move from the average (nominal) coefficient value. Protecting against the full range of a Gaussian distribution, in fact, would involve setting the coefficient half-value to infinity! This could be avoided by redefining the half-values to be one standard deviation from the coefficient mean, for example. Also, the current mode of protection might actually

be warranted when the distribution of uncertainty is weighted more heavily at its tails. This would be the case if we assumed a binary distribution for the uncertainty. In our example, part of the uncertainty in task accomplishment was due to our not knowing whether targets actually existed at the predefined target locations. The probability distribution might then be a 70% chance that the target exists and full reward is gained, and a 30% chance that it does not and zero reward is gained. If the realizations are weighted on the bounds of our range, it is logical that we protect on those bounds as well.

A final observation is that, if we are uncertain of the true value of the coefficient, how likely is it that we are certain of the range in which it lies? The Bertsimas-Sim method can still prove useful even if the precise coefficient boundaries are not known, and we acknowledge that modeling would not be possible without the use of assumptions.

6.3 Recommendations on Modification

We now make recommendations for changes in the UAVMPP formulation, in both its application and its composition. Our suggested changes apply to the Bertsimas-Sim portion in specific, as well as to the formulation in whole. Among our general formulation comments, we feel that the formulation would be improved through the use of a better method for measuring range. At present, we measure range by summing up the service and travel times incurred by the UAVs. This method prevents the formulation from considering loiter time, where the plane is airborne but not actively engaged in a task or flying between tasks. While it is probably valid that loitering expends less fuel and therefore less of the range capacity, our assumption of no fuel consumption removes some realism from the formulation.

Another possible modification to the formulation would be that of allowing tasks to be performed multiple times at targets, and using a reward function that decreases with each additional task. This modification would add considerable complexity to the problem, as it would involve the use of adding a separate decision variable for

every subsequent performance of the task, but would also add more flexibility to the formulation and might provide added insight. This change could be coupled with an additional Battle Damage Assessment (BDA) task, and a closed-loop formulation that constantly adjusts its plan with new information might be developed.

We encountered a fair amount of additional formulation complexity as a result of modeling two task types. Another avenue of modification would be to find a way of simplifying the formulation while still providing this valuable capability. A possible alternative would be that of modeling the tasks as separate targets located in the same geographic locations. We opted against this method because it added to the input data required for the formulation, but this method might prove preferable if it results in a more tractable formulation.

We turn now to the handling of uncertainty. Another possible modification to the formulation would be to model the possibility that UAVs are lost in the course of operations. UAVs could be lost as a result of an enemy attack or aircraft malfunction, and this possibility could be specifically protected by the formulation.

Regarding the Bertsimas-Sim formulation specifically, we suggest that it be only used for constraints that have multiple uncertain coefficients. As mentioned earlier, the formulation reduces to adding or subtracting a fixed amount from the coefficient in the cases where only one uncertain coefficient exists. We also suggest that, if the Bertsimas-Sim formulation is applied to the objective function, that an estimate for solution reward be derived by some other means. As revealed in the testing, protected objective functions tended to be overly pessimistic and ceased from being good estimates for plan reward. Such estimates, if desired, could be gained by multiplying the found solution with the average task reward values, or could be obtained through the use of simulation.

We also suggest the use of Lagrangians as demonstrated in Experiment 7. The Lagrangian additions to the objective function cause the formulation to discriminate between solutions on the basis of slack, but also open the opportunity for constraints to be weighted differently. Constraints that are deemed more important, for instance, can be weighted more heavily in the objective function, and therefore receive

protection above that which is already provided by the Bertsimas-Sim formulation. Furthermore, the use of Lagrangians in the objective function resulted in solutions that behaved much more consistently with added protection. Previously, because the formulation did not differentiate between solutions on the basis of slack, it was possible for solutions with added protection to perform worse with regard to the metrics. This could prove to be a serious problem for someone deciding whether or not to use protection.

Another recommendation for the Bertsimas-Sim formulation is that it be applied on a constraint by constraint basis. This is possible in the original formulation presented in [12], but not in our formulation because of our use of the *prot* parameter. While our intention in specifying a blanket level of protection was to simplify the options for protection, doing so did not provide the control over the level of protection that one might seek. In our case, the objective function had many uncertain coefficients, but only a relatively small percent of them were multiplied by nonzero decision variables in the solutions. With protection specified as a fraction of the total number of uncertain coefficients in the constraints, the objective function reached its maximum protection much earlier in the protection levels than did the constraints. Applying protection on a constraint by constraint basis would allow the protection ranges to be aligned in a more desirable manner.

In a more fundamental change, a different way of characterizing the protection level might also be helpful. At present, both Γ and *prot* still define the level of protection as the number (or percent) of uncertain coefficients that assume their worst-case values in each constraint. This way of defining uncertainty might not be intuitive for a decision-maker, who is likely to be interested in a gauge that relates more directly to solution performance. One alternative would be to relate the protection level to its tradeoff with objective function value. Protection, then, could be measured by the percent of objective function value that must be foregone as a result of the protection. Marla presents a formulation that relates protection to foregone objective function value in [31]. As we have seen, however, objective function value is not a good estimate of plan value when the objective function itself is protected, and

this modification should be coupled with an alternative means of protection for the objective function itself.

Another means of selecting protection could come through simulation and through a unified metric. We decided against a unified metric for the UAVMPP because we were not prepared to make the assumptions necessary to rank and weight the current metrics. The operator of a particular system, however, being intimately involved with the problem, would be more apt to make such a decision. Armed with a unified metric, it would be simple to compare the simulation results of different protection levels, and protection could even be optimized with respect to this metric.

A final suggestion on the modification of the formulation is to find a way to weight violations on the basis of time. As was revealed in the output graphs of the Departure Times, the formulation in its present state does not differentiate violations on the basis of their time of occurrence. The Departure Time for a particular violation type, for example, could decrease and increase in a seemingly arbitrary manner. It might be possible to apply protection in a way that safeguards more against departures that occur early in the plan than those that occur later.

6.4 Future Work

In addition to the modifications suggested in the previous section, we mention several other opportunities for continued work in this field. A considerable amount of testing might still be performed for the modified formulation we posed in Experiment 7, for instance. It would be helpful to fully characterize the scenarios in which protection caused all metrics to improve. We also intended on testing the formulation on different assumptions of the distribution of uncertainty. It is possible that distributions other than the uniform distribution might govern data uncertainty, and it is likewise possible that the formulation would perform differently under these different assumptions.

In addition, this research would be strengthened if it was accompanied with testing of the formulation under different modes of protection. It would be interesting to see, for instance, how the Bertsimas-Sim solutions fare against the simple method of

adjusting the coefficients directly.

We also suggest that other methods or formulations of the UAVMPP be considered. Heuristics, for instance, might be found to produce solutions that do not give up excessive amounts of optimality. While current heuristic methods do not incorporate uncertainty, it might be possible to couple them with a robust optimization technique. Doing so could provide benefits in terms of tractability and the size of problems that could be handled.

Finally, this research into developing and evaluating a robust mid-level planner could be accompanied by the development of robust high-level and low-level planners. The Masters Thesis, 'Robust, Theater-Level, Effects-Based, Mission Planning for UAVs', by Corban Bryant, presents a robust high-level planner [18]. In addition, Draper Laboratory has developed several low-level path planners (among which are [32] [23]). These high- and low-level planners could be integrated with our mid-level planner into one system that would automate and optimize the entire targeting process.

Appendix A

"Value of Information" Formulation

This formulation of the UAVMPP is modified to incorporate the "value of information" using "prep" variable. "prep" is turned on when an ISR is performed at a target. It acts to remove the uncertainty in a subsequent strike of the target. This formulation seeks to investigate the question of how added information concerning a target should be valued.

$$
\begin{aligned}
\max \sum_{k\in K}\sum_{i\in N} C_{ik}y_{ik} \;-\; & {}_2\Gamma_1\ {}_2\theta_1 - \sum_{i\in N}\sum_{k\in K} {}_2\rho_{ik} \\
\text{s.t} \sum_{k\in\{Strike\}} y_{ik} \;&\leq\; 1 \quad \forall i \in T \\
\sum_{k\in\{ISR\}} y_{ik} \;&\leq\; 1 \quad \forall i \in T \\
\sum_{j\in N} x_{0jk} \;&=\; 1 \quad \forall k \in K \\
\sum_{i\in N} x_{i,n,k} \;&=\; 1 \quad \forall k \in K \\
\sum_{i\in N} x_{ijk} \;&=\; \sum_{i\in N} x_{jik} \;=\; y_{jk} \quad \forall k \in K,\ \forall j \in T \\
w_{ik} + S_{ik} + D_{ijk} + {}_8\Gamma_{ijk}\ {}_8\theta_{ijk} + \sum_{u\in[1,2]} {}_8\rho_{uijk} \;&\leq\; w_{jk} + (1 - x_{ijk})M \quad \forall k \in K,\ \forall i,j \in N \\
A_i y_{ik} + {}_9\Gamma_{ki}\ {}_9\theta_{ki} + {}_9\rho_{ki} \;&\leq\; w_{ik} \quad \forall k \in K,\ \forall i \in T \\
w_{ik} + {}_{10}\Gamma_{ki}\ {}_{10}\theta_{ki} + {}_{10}\rho_{ki} \;&\leq\; B_i y_{ik} \quad \forall k \in K,\ \forall i \in T \\
w_{n,k} + {}_{11}\Gamma_k\ {}_{11}\theta_k + {}_{11}\rho_k \;&\leq\; R_k \quad \forall k \in K \\
\sum_{k\in K} y_{ik} - 1 \;&\leq\; prep_i \quad \forall i \in T \\
prep_i \;&\leq\; \sum_{r\in ISR} y_{ir} \quad \forall i \in T \\
prep_i \;&\leq\; \sum_{s\in Strike} y_{is} \quad \forall i \in T \\
w_{ir} + S_{ir}y_{ir} - M(1 - y_{is}) + {}_{12}\Gamma_{irs}\ {}_{12}\theta_{irs} + {}_{12}\rho_{irs} \;&\leq\; w_{is} \quad \forall i \in T, \forall r \in ISR,\ \forall s \in Strike
\end{aligned}
$$

$$
\begin{aligned}
\hat{C}_{ik} y_{ik} &\leq {}_2\theta_1 + {}_2\rho_{ik} \quad \forall i \in N \; \forall k \in ISR \\
\hat{C}_{ik} y_{ik} - M(prep_i) &\leq {}_2\theta_1 + {}_2\rho_{ik} \quad \forall i \in N \; \forall k \in Strike \\
\hat{S}_{ik} y_{ik} &\leq {}_8\theta_{ijk} + {}_8\rho_{1ijk} \quad \forall i,j \in N \; \forall k \in ISR \\
\hat{S}_{ik} y_{ik} - M(prep_i) &\leq {}_8\theta_{ijk} + {}_8\rho_{1ijk} \quad \forall i,j \in N \; \forall k \in Strike \\
\hat{D}_{ijk} x_{ijk} &\leq {}_8\theta_{ijk} + {}_8\rho_{2ijk} \quad \forall i,j \in N \;\; \forall k \in K \\
\hat{A}_i y_{ik} &\leq {}_9\theta_{ki} + {}_9\rho_{ki} \quad \forall i \in N \; \forall k \in ISR \\
\hat{A}_i y_{ik} - M(prep_i) &\leq {}_9\theta_{ki} + {}_9\rho_{ki} \quad \forall i \in N \; \forall k \in Strike \\
\hat{B}_i y_{ik} &\leq {}_{10}\theta_{ki} + {}_{10}\rho_{ki} \quad \forall i \in N \; \forall k \in ISR \\
\hat{B}_i y_{ik} - M(prep_i) &\leq {}_{10}\theta_{ki} + {}_{10}\rho_{ki} \quad \forall i \in N \; \forall k \in Strike \\
\hat{R}_k &\leq {}_{11}\theta_k + {}_{11}\rho_k \quad \forall k \in K \\
\hat{S}_{ir} y_{ir} &\leq {}_{12}\theta_{irs} + {}_{12}\rho_{irs} \; \forall i \in T, \; \forall r \in ISR \, \forall s \in Strike \\
x_{ijk} &\in \{0,1\} \quad \forall k \in K, \; \forall i,j \in N \\
y_{ik} &\in \{0,1\} \quad \forall k \in K, \; \forall i \in N \\
w_{ik} &\geq 0 \quad \forall i \in N, \; \forall k \in K \\
\rho, \theta &\geq 0 \\
prep_i &\geq 0 \quad \forall i \in N
\end{aligned}
$$

Appendix B

"ISR before Strike" Formulation

This formulation of the UAVMPP is modified to ensure ISR of a target before strike. The constraint was relaxed in future formulations.

$$
\begin{aligned}
\min \sum_{k\in K}\sum_{i\in N} C_{ik}y_{ik} \;+\; {}_2\Gamma_1\ {}_2\theta_1 + \sum_{i\in N}\sum_{k\in K} {}_2\rho_{ik} & \\
\text{s.t} \sum_{k\in\{Strike\}} y_{ik} \;\le\; 1 \quad \forall i \in T & \\
\sum_{k\in\{ISR\}} y_{ik} \;\le\; 1 \quad \forall i \in T & \\
\sum_{j\in\Delta^+(0)} x_{0jk} \;=\; 1 \quad \forall k \in K & \\
\sum_{i\in\Delta^-(n+1)} x_{i,n+1,k} \;=\; 1 \quad \forall k \in K & \\
\sum_{i\in\Delta^-(j)} x_{ijk} \;=\; \sum_{i\in\Delta^+(j)} x_{jik} \;=\; y_{jk} \quad \forall k \in K,\ \forall j \in T & \\
w_{ik} + S_{ik}y_{ik} + D_{ijk}x_{ijk} + {}_8\Gamma_{ijk}\ {}_8\theta_{ijk} + \sum_{u\in[1,2]} {}_8\rho_{uijk} \;\le\; w_{jk} + (1 - x_{ijk})M_{ij} \quad \forall k \in K,\ \forall i,j \in N & \\
A_iy_{ik} + {}_9\Gamma_{ki}\ {}_9\theta_{ki} + {}_9\rho_{ki} \;\le\; w_{ik} \quad \forall k \in K,\ \forall i \in T & \\
w_{ik} + {}_{10}\Gamma_{ki}\ {}_{10}\theta_{ki} + {}_{10}\rho_{ki} \;\le\; B_iy_{ik} \quad \forall k \in K,\ \forall i \in T & \\
w_{n+1,k} + {}_{11}\Gamma_k\ {}_{11}\theta_k + {}_{11}\rho_k \;\le\; R_k \quad \forall k \in K & \\
w_{ir} + S_{ir}y_{ir} + {}_{12}\Gamma_{irs}\ {}_{12}\theta_{irs} + {}_{12}\rho_{irs} \;\le\; w_{is} \quad \forall i \in T, \forall r \in \{ISR\},\ \forall s \in \{Strike\} & \\
\sum_{s\in Strike} y_{is} \;\le\; \sum_{r\in ISR} y_{ir} \quad \forall i \in T & \\
\hat{C}_{ik}y_{ik} \;\le\; {}_2\theta_1 + {}_2\rho_{ik} \quad \forall i \in N\ \forall k \in K & \\
\hat{S}_{ik}y_{ik} \;\le\; {}_8\theta_{ijk} + {}_8\rho_{1ijk} \quad \forall i,j \in N\ \forall k \in K & \\
\hat{D}_{ijk}x_{ijk} \;\le\; {}_8\theta_{ijk} + {}_8\rho_{2ijk} \quad \forall i,j \in N\ \ \forall k \in K & \\
\hat{A}_iy_{ik} \;\le\; {}_9\theta_{ki} + {}_9\rho_{ki} \quad \forall i \in N\ \forall k \in K & \\
\hat{B}_iy_{ik} \;\le\; {}_{10}\theta_{ki} + {}_{10}\rho_{ki} \quad \forall i \in N\ \forall k \in K & \\
\hat{R}_k \;\le\; {}_{11}\theta_k + {}_{11}\rho_k \quad \forall k \in K & \\
\hat{S}_{ir}y_{ir} \;\le\; {}_{12}\theta_{irs} + {}_{12}\rho_{irs} \quad \forall i \in T\ \forall r \in \{ISR\} & \\
\qquad\qquad \forall s \in \{Strike\} & \\
x_{ijk} \;\in\; \{0,1\} \quad \forall k \in K,\ \forall i,j \in N &
\end{aligned}
$$

$$
\begin{aligned}
y_{ik} &\in \{0,1\} \quad \forall k \in K,\ \forall i \in N \\
w_{ik} &\geq 0 \quad \forall i \in N,\ \forall k \in K \\
\rho &\geq 0 \\
\theta &\geq 0
\end{aligned}
$$

Bibliography

[1] Richa Agarwal, Ravindra K. Ahuja, Gilbert Laporte, and Zuo-Jun Shen. A composite very large-scale neighborhood search algorithm for the Vehicle Routing Problem, 2003.

[2] Amr Ahmed, Abhilash Patel, Tom Brown, MyungJoo Ham, Myeong-Wuk Jang, and Gul Agha. Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism. In *The International Conference of Integration of Knowledge Intensive Multi-Agent Systems KIMAS 05: Modeling, Evolutions and Engineering, pp. 311-317, April 18 - 21*, 2005.

[3] R. K. Ahuja, A. Kumar, and K. Jha. Exact and heuristic methods for the weapon target assignment problem. *Social Science Research Network Electronic Library*, 2003.

[4] US Air Force. Air Force Instruction 13-1 AOC, August 2005. http://www.e-publishing.af.mil/pubs/publist.asp?puborg=AF&series=13.

[5] Rafael Aleman. Multiperiod task assignment problem for unmanned aerial vehicles. Technical report, Wright State University.

[6] G. B. Alvarenga, G. R. Mateus, and G. de Tomi. Finding near optimal solutions for vehicle routing problems with time windows using hybrid genetic algorithm. citeseer.ist.psu.edu/646969.html.

[7] Guilherme Bastos Alvarenga and Geraldo Robson Mateus. A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows. *Hybrid Intelligent Systems*, 2004.

[8] Jonathan F. Bard, George Kontoravdis, and Gang Yu. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36(2):250–269, May 2002.

[9] R. Bent and P. Hentenryck. Dynamic vehicle routing with stochastic requests, 2003. citeseer.ist.psu.edu/bent03dynamic.html.

[10] R. Bent and P. Hentenryck. Scenario-based planning for partially dynamic routing with stochastic customers, 2003. citeseer.ist.psu.edu/640332.html.

[11] Russell Bent and Pascal Van Hentenryck. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4):515–530, 2004. citeseer.ist.psu.edu/brent01twostage.html.

[12] Dimitris Bertsimas and Melvyn Sim. The Price of Robustness. Technical report, Operations Research Center, Massachusetts Institute of Technology, 2002.

[13] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, Massachusetts, 1997.

[14] J. Bramel and D. Simchi-Levi. On the effectiveness of set covering formulations for the vehicle routing problem with time windows. *Operations Research*, 45:295–301, 1997. citeseer.ist.psu.edu/138965.html.

[15] O. Bräysy and M. Gendreau. Tabu search heuristics for the Vehicle Routing Problem with Time Windows. In *TOP*, pages 211–238, 2002.

[16] Olli Bräysy. Vehicle Routing Problem with Time Windows, part i: Route construction and local search algorithms.

[17] DoD News Briefing. Background Briefing on Unmanned Aerial Vehicles, October 2001. http://www.fas.org/irp/program/collect/uav_103101.html.

[18] Corban H. Bryant. Robust, theater-level, effects-based, mission planning for uavs. Master's thesis, Massachusetts Institute of Technology, 2006.

[19] Ang Juay Chin, Ho Wee Kit, and Andrew Lim. A new GA approach for the Vehicle Routing Problem. In *11th IEEE Conference on Tools with Artificial Intelligence TAI 99*, November 1999.

[20] US Joint Forces Command. Joint Publication 3-60. Joint Doctrine for Targeting, January 2002. http://www.dtic.mil/doctrine/jpoperationsseriespubs.htm.

[21] W. Cook and J. Rich. A parallel cutting-plane algorithm for the vehicle routing problem with time windows, 1999. citeseer.ist.psu.edu/cook99parallel.html.

[22] J.-F. Cordeau, Guy Desaulniers, Jacques Desrosiers, Marius M. Solomon, and Francois Soumis. The VRP with Time Windows. In *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications, 2000.

[23] Christopher W. Devers. Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes. *Journal of Guidance Control and Dynamics, AIAA*, 29(2), 2006.

[24] A. W. Drake. *Observation of a Markov Process Through a Noisy Channel.* PhD thesis, Massachusetts Institute of Technology, 1962.

[25] Matthew Earl and Raffaello D'Andrea. A decomposition approach to multi-vehicle cooperative control, 2005. http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cs/0504081.

[26] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 63–76. McGraw-Hill, 1999. citeseer.ist.psu.edu/article/gambardella99macsvrptw.html.

[27] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research.* McGraw Hill, New York, New York, seventh edition, 2001.

[28] Paul Juell, Amal Perera, and Kendall E. Nygard. Application of a genetic algorithm to improve an existing solution for the generalized assignment problem. In *16th International Conference on Computer Applications in Industry and Engineering (CAINE 03)*, 2003.

[29] Kai Kwong Lau, M. J. Kumar, and N. R. Achuthan. Parallel implementation of branch and bound algorithm for solving vehicle routing problem on nows. In *ISPAN*, pages 247–253, 1997.

[30] Andrew Lim and Fan Wang. A smoothed dynamic tabu search embedded GRASP for m-VRPTW. In *16th IEEE Conference on Tools with Artificial Intelligence ACTAI 2004*, 2004.

[31] Lavanya Marla. Robust Optimization for Large-scale Network-based Resource Allocation Problems, 2006. CEE-PhD General Examination Working Research Paper, Department of Civil and Environmental Engineering, MIT.

[32] S.D. McKeever. Path planning for an autonomous vehicle. Master's thesis, Massachusetts Institute of Technology, 2000.

[33] Noor Hasnah Moin. Hybrid genetic algorithms for Vehicle Routing Problems with Time Windows. *International Journal of the Computer, the Internet and Management*, 10(1), January 2002.

[34] R. Montemanni, L. Gambardella, A. Rizzoli, and A. Donati. A new algorithm for a dynamic Vehicle Routing Problem based on ant colony system, 2003. citeseer.ist.psu.edu/article/montemanni03new.html.

[35] Beatrice Ombuki, Brian J. Ross, and Franklin Hanshar. Multi-objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. Technical report, Brock University, 2004.

[36] David Pisinger. *Transportation and Cargo Loading* class lecture slides. DIKU, University of Copenhagen. http://www.diku.dk/undervisning/2005e/426/.

[37] L. Rousseau, M. Gendreau, and G. Pesant. Using constraint-based operators with variable neighborhood search to solve the vehicle routing problem with time windows. citeseer.ist.psu.edu/rousseau99using.html.

[38] Corey Schumacher and Phillip Chandler. Constrained optimization for uav task assignment. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 1994.

[39] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

[40] K. C. Tan, T. H. Lee, K. Ou, and L. H. Lee. A messy genetic algorithm for the Vehicle Routing Problem with time window constraints. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 679 – 686, 2001.

[41] P. Toth and D. Vigo. *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications, 2002.

[42] Kirk A. Yost. *Solution of Large-Scale Allocation Problems with Partially Observable Outcomes.* PhD thesis, Naval Postgraduate School, 1998.

[43] Eric J. Zarybnisky. Allocation of air resources against an intelligent adversary. Master's thesis, Massachusetts Institute of Technology, 2003.